

# Exposé

*„Verteilung von Updates in mobilen Ad Hoc Netzwerken“*

Manuel Scholz

Betreuer: Prof. Dr. H. Schweppe,

Institut für Informatik

Freie Universität Berlin

[mscholz@inf.fu-berlin.de](mailto:mscholz@inf.fu-berlin.de)

16.10.2003

## 1. Einleitung

Gerade in den letzten Jahren ist durch die zunehmende Verkleinerung der Computer (Laptop, PDA) und des immensen Wachstums der drahtlosen Datenübertragung (WLAN, Bluetooth) das Thema Mobilität in den Vordergrund gerückt. Aufgrund dieser Tatsache findet sich gerade im Bereich der mobilen Datenverarbeitung eine Vielzahl an neuen Möglichkeiten. Ein aktuelles Beispiel sind die sogenannten "mobilen Ad Hoc Netzwerke". Als sogenannte MANETS werden dezentrale und unstrukturierte P2P Netzwerke, die sich per Funkübertragung spontan zwischen mehreren mobilen Rechnern bilden können, bezeichnet. Eine Besonderheit ist dabei die durch die Mobilität hervorgerufene hohe Dynamik. Durch die ständige Aktivität der einzelnen Netzteilnehmer hat das MANET keine feste Struktur (Topologie), sondern ändert diese kontinuierlich. Somit lassen sich viele bekannte Methoden aus den bisher untersuchten (statischen) Netzwerken oft gar nicht oder nur bedingt umsetzen. Aus diesem Grund gibt es in diesem Bereich viele bisher wenig erforschte Problemstellungen, wie z.B. die Verteilung und die Synchronisation von Objekten oder das Routing innerhalb eines MANETS.

Im Rahmen des Schwerpunktprogrammes 1140 der deutschen Forschungsgesellschaft setzt sich das CoCo/Da (Cooperation & Communication / Data) Projekt der Arbeitsgruppe "Datenbanken und Informationssysteme" mit diesem Thema auseinander. Ziel des Projektes ist die Entwicklung einer Basissoftware (Middleware) für vernetzte mobile

Systeme. Als konkretes Szenario soll dabei ein Touristeninformationssystem dienen. In diesem Szenario bilden sich Ad Hoc Netzwerke zwischen Rechnern an touristischen Informationspunkten und den einzelnen Touristen und ihren mobilen Rechnern.

Im folgenden soll konkret das Problem der Verteilung von Softwareupdates in diesem Szenario (und in MANETS im allgemeinen) betrachtet werden.

## **2. Problemstellung**

Die effiziente Verteilung von Updates in einem gegebenen Netz ist bereits in vielen Teilgebieten untersucht worden [1,6,2]. Aufgaben wie die Updates von Webseiten-spiegelungen, DNS-Servern oder auch die Aktualisierung von verteilten Datenbanken sind typische Problemstellungen der Updateverteilung aus der Praxis. Grundsätzlich gibt es zur Lösung dieses Problem zwei Ansätze: Den push-basierten Ansatz, in dem die Updates aktiv verteilt werden und den pull-basierten Ansatz bei dem die Updateinformation erfragt und ggf. geholt wird. Die Wahl des Ansatzes hängt im wesentlichen von der Frequenz der Updates im Netzwerk ab.

Neben vielen anderen Ansätzen findet bei den push-basierten Algorithmen besonders die Klasse der epidemischen Algorithmen [3,4] Anwendung, da sie nur eine geringe Netzlast erzeugt und so gut wie keine Voraussetzung an die Topologie des Netzwerkes stellt. Aufgrund dieser Eigenschaften soll dieser Ansatz ebenfalls bei der Verbreitung von Updates in MANETS untersucht werden.

## **3. Verteilung von Softwareupdates in mobilen Ad Hoc Netzwerken**

### **3.1 Voraussetzungen**

Im Gegensatz zu den bereits beschriebenen (meist) statischen Netzwerken sind in einem mobilen Ad Hoc Netzwerk andere Voraussetzungen gegeben:

- *Unbekannte Topologie*  
Durch die hohe Dynamik des MANETS ändert sich die Topologie kontinuierlich und kann somit nicht genutzt werden.
- *Begrenzte Energie*  
Die mobilen Netzknoten (PDA, Handy, Laptop) besitzen nur eine begrenzte Menge an Energie, wodurch die Nutzung zeitlich beschränkt ist und somit minimiert werden muss.
- *Geringe Bandbreite*  
Die mobilen Netzknoten haben im Vergleich zu anderen Netzwerken nur eine geringe Bandbreite, was nur eine geringe Netzlast im MANET ermöglicht.

### **3.2 Ziele**

Ziel ist es, die Verbreitung der Updates möglichst effizient zu gestalten. Dabei ist die wichtigste Ergebnisdimension die Anzahl der erreichten Netzknoten. Weitere zu betrachtende Dimensionen sind die Zeit, die Netzlast und der Energieverbrauch. Die Anzahl der erreichten Netzknoten ist also unter bestimmten Randbedingungen (Zeit, Netzlast, Energieverbrauch) zu maximieren.

### **3.3 Algorithmen**

Mit den gegebenen Voraussetzungen eines mobilen Ad Hoc Netzwerkes und dem Ziel die Verbreitung der Updates so effizient wie möglich umzusetzen müssen nun einzelne Verteilungsalgorithmen bezüglich dieser Kriterien untersucht werden. Im folgenden werden die Ansätze der zu untersuchenden Algorithmen kurz vorgestellt.

Vorab ist festzuhalten, dass jeder Netzknoten eine Liste mit den letzten von ihm gesehenen Updates (für jedes vorhandene Programm) besitzt, so dass die Algorithmen zwecks Vermeidung unnötiger Verbreitung auf diese Liste zurückgreifen können.

- *Simple Flooding*  
Der simple Flooding Algorithmus verteilt die Daten im Netz durch einfaches fluten, d.h., dass jeder Netzknoten das Update an alle Nachbarn weiterschickt.
- *TTL Flooding*  
TTL Flooding verteilt die Daten ebenfalls durch einfaches Fluten des Netzes, jedoch ist die Reichweite durch einen TTL (Time To Live) Wert begrenzt. Trifft das Update auf ein Programm, welches dieses Update benötigt, so wird der TTL Wert wieder erneuert.
- *Random Walker* [5]  
Im Gegensatz zu der Verteilung durch ein Fluten des Netzwerkes, versendet dieser Algorithmus  $n$  Random Walker, welche sich zufällig im Netz bewegen und dabei das Update verteilen. Trifft ein Random Walker auf einen Knoten, welcher das Update bereits gesehen hat, so stoppt er mit einer gegebenen Wahrscheinlichkeit.
- *Epidemic Distribution* [3,4]  
Der Epidemic Distribution Algorithmus verteilt das Update nicht an alle Nachbarn, sondern nur an eine Teilmenge. Die Teilmenge wird durch eine gegebene Wahrscheinlichkeit bestimmt.

Erweiterungen der Algorithmen:

- *Informationsanfrage*  
Vor der eigentlichen Verteilung der Updates wird an jeden Nachbarn des verteilenden Knotens eine Anfrage über dessen Update Status gestellt. Die Verteilung der Updates wird der Rückmeldung entsprechend angepasst.
- *Replikation des Update Status*

Jeder Netzknoten verbreitet die Information seiner benötigten Updates in einem bestimmten Umkreis.

- *Wiederholung der Verteilung in Intervallen*  
Die Verteilung der Updates wird von bestimmten Knoten in Intervallen wiederholt, um Knoten zu erreichen, welche bei der letzten Verteilung nicht erreichbar waren.
- *Pull-basierte Verteilung*  
Erweiterung der Algorithmen um pull-basierte Ansätze, wie z.B. das Abfragen nach Updates der Nachbarn durch einen neu hinzugekommenen Knoten.

Viele der Algorithmen und Erweiterungen sollen nicht nur getrennt voneinander betrachtet werden, sondern es sollen auch sinnvolle Kombinationen der einzelnen Ansätze untersucht werden.

### **3.4 Experimentelle Untersuchung**

Die experimentelle Untersuchung der Algorithmen für einzelne Netzwerkszenarien soll anhand einer Simulation durch OMNeT++ durchgeführt werden. Ziel dieser Untersuchung ist es jeweils den idealen Algorithmus für ein spezielles Netzwerkszenario (Clusterung, Größe, Dichte) zu finden. Diese Ergebnisse sollen eine adaptive Middleware für mobile Ad Hoc Netzwerke ermöglichen, welche sich bezüglich der Verteilung an das jeweilige Szenario anpassen kann.

### **3.5 Weitere Planung**

Nachdem in der ersten Untersuchung der Schwerpunkte auf die eigentlich Verteilung der Daten gelegt wurde, soll in weiteren Untersuchungen besonders die Randbedingungen des gesamten Szenarios betrachtet werden. So lässt sich z.B. Durch die Verwendung von unterschiedlichen Updateverfahren (komplett, partiell) zusätzlichen starken Einfluss auf die Effizienz der Verteilung nehmen. Weiterhin soll betrachtet werden, in wie weit sich die Verteilungsalgorithmen dieses speziellen Szenarios auf allgemeinere Szenarien anwenden lassen.

## **7. Literatur**

- [1] G. Coulouris, J. Dollimore, T. Kindberg. *Distributed Systems Concepts and Design*. Pearson Education Ltd, 2002
- [2] A. Datta, M. Hauswirth, Karl Aberer. Updates in Highly Unreliable, Replicated Peer-to-Peer Systems. *In the proceedings of the 23rd International Conference on Distributed Computing Systems, ICDCS2003*. 2002
- [3] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, D. Terry. Epidemic Algorithms for Replicated Database Maintenance. *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*. 1987
- [4] R. Karp, C. Schindelhauer, S. Shenker, B. Vöcking. Randomized Rumor Spreading. *IEEE Symposium on Foundations of Computer Science*. 2000
- [5] Q. Lv, P. Cao, E. Cohen, K. Li, S. Shenker. Search and replication in unstructured peer-to-peer networks. *Proceedings of the 16<sup>th</sup> annual ACM International Conference on supercomputing*. 2002
- [6] A. Tanenbaum, M. van Steen. *Verteilte Systeme, Grundlagen von Paradigmen*. , Pearson Education Deutschland, 2002