

XCast-Kommunikation im Virtual Shared Information Space

Florian Reuter, Jochen Koberstein und Norbert Luttenberger
Institut für Informatik und praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Christian-Albrechts-Platz 4, D-24098 Kiel, Germany
{flr|jko|nl}@informatik.uni-kiel.de

Zusammenfassung Der Einsatz eines Client/Server-Kooperationsparadigmas ist für mobile Ad-Hoc Netzwerke problematisch, da nicht von der ständigen Verfügbarkeit und Erreichbarkeit von Servern ausgegangen werden kann. In diesem Artikel wird daher das serverlose Kooperationsparadigma des *virtual shared information space* vorgestellt, welches durch eine XML-basierte datenzentrierte Kommunikationsform namens XCast realisiert wird. Im zweiten Teil des Artikels wird eine Anwendung vorgestellt, der auf der Basis von XCast einen *virtual shared information space* für eine endliche Informationsmenge etabliert. Die Leistungsbewertung zeigt, dass die Anzahl der dazu nötigen Pakete von einem sog. Dokumentengrenzwert abhängt. Die in der Literatur etablierte Vorstellung, dass eine datenzentrierte Kommunikation innerhalb von mobilen Ad-Hoc Netzen vorteilhaft ist, konnte mit Hilfe dieses Grenzwertes mathematisch vorhergesagt und simulativ bestätigt werden.

1 Einleitung

Das heute am weitesten verbreitete Kooperationsparadigma ist das Client/Server-Paradigma, zu dem sog. request/response-Kommunikationsprotokolle gehören. In mobilen selbstorganisierenden — d.h. infrastrukturlosen — Umgebungen stößt dieses Kooperationsparadigma erkennbar an seine Grenzen, da es de facto von der Erreichbarkeit bestimmter Dienste in einer mehr oder weniger statischen Infrastruktur ausgeht. Diese ist jedoch unter den Bedingungen der Mobilität und dem Medium Funk nicht dauerhaft oder nur um den Preis hoher Kommunikationskosten gegeben. Selbst mit den auf unterschiedlichen Formen der Dienstbeschreibung aufbauenden Dienstfindungsmechanismen verbessert sich die Situation nicht grundlegend, da hier der Client/Server-Kooperation nur ein weiterer Mechanismus quasi “vorgesaltet” wird.

In diesem Beitrag wird deshalb untersucht, welche Konsequenzen sich für die Kommunikation in mobilen selbstorganisierenden Umgebungen ergeben, wenn Anwendungssysteme für gemeinsam operierende “Schwärme” von mobilen funkvernetzten Systemen nicht nach Client/Server-Manier miteinander kooperieren, sondern durch gleichberechtigte lesende und schreibende Zugriffe auf einen *virtual shared information space* (VSIS). Die aus diesem Kooperationsparadigma resultierende Kommunikationsform bezeichnen wir als datenzentrierte Kommunikation, weil dabei Daten nur dann übertragen werden, wenn sie neue Information enthalten. Im Umfeld der mobilen selbstorganisierenden funkvernetzten Systeme wird bewusst davon ausgegangen, dass nicht immer

alle Mitglieder des Schwarms vollständig funktionsfähig und funktechnisch erreichbar sind.

Die Etablierung von VSIS ist immer da sinnvoll, wo eine endliche Informationsmenge von allen Kommunikationsteilnehmern gemeinsam verarbeitet werden muss. Endliche Informationsmengen sind, im Gegensatz zu unendlichen Informationsmengen, in ihrer Kardinalität beschränkt. Beispiele für endliche Informationsmengen sind Datenbestände, Beispiele für unendliche Informationen sind kontinuierliche Messwerte oder auch Multimedia Streams. Einige der unendlichen Informationen können allerdings in praktischen Anwendungen durch eine diskrete Abtastung in endliche überführt werden oder sind innerhalb eines Zeitintervalls konstant und damit endlich. Ein VSIS, welcher die Temperaturen aller Räume eines Gebäudes speichert und somit die Grundlage für z.B. Feuerbekämpfungsmaßnahmen bildet, kann die Menge der Temperaturen durch Diskretisierung in 1°C Schritte im Bereich von -5°C bis 100°C endlich machen, wodurch aufgrund der üblicherweise geringen Temperaturschwankungen innerhalb von Räumen eine endliche Informationsmenge entsteht.

Ein VSIS welcher die Verkehrsinformationen: Staus, Umleitungen, ... einer Stadt speichert, ist sicherlich in einem Zeitintervall endlich, d.h. an einem Vormittag treten z.B. nur endlich viele Staus auf.

In dieser Arbeit wird zur Demonstration und Evaluation ein einfacher VSIS etabliert, dem folgende Applikationsidee zugrunde liegt: Alle Teilnehmer einer Konferenz sind mit einem elektronischen Adressbuch ausgerüstet, die untereinander über IEEE 802.11 Hardware kommunizieren können. Ziel sei es, eine Liste aller Teilnehmer zu etablieren, ohne dafür einen zentralen Server zu nutzen.

Zur Etablierung eines VSIS schlagen wir einen neuartigen XML basierten Kommunikationsmechanismus vor, den wir als XCast bezeichnen. Im folgenden Abschnitt dieses Artikels werden XCast und die zugehörigen Designprinzipien vorgestellt. Im dritten Abschnitt wird dann gezeigt, wie man aufbauend auf XCast einen VSIS für endliche Informationen etabliert. Anhand des Teilnehmerlisten VSIS wird mit Hilfe einer durch Simulationsergebnisse bestätigten mathematischen Analyse gezeigt, mit welchem Aufwand und in welcher Zeit endliche Informationen innerhalb eines Netzes verteilt werden können.

2 Das datenzentrierte Kooperationsparadigma des Virtual Shared Information Space

2.1 Die XCast-Kommunikationstechnik

Bei datenzentrierter Kommunikation ergibt sich die Rolle eines Kommunikationspartners allein daraus, wel-

che Daten dieser zur Verfügung hat. Kommunikationspartner mit mehr Informationen verbreiten diese an Kommunikationspartner mit weniger Informationen, bis sich ein Gleichgewicht einstellt, bei dem idealerweise alle Kommunikationspartner dieselben Informationen besitzen.

Ein Kommunikationsteilnehmer j besitzt einen lokalen Informationsspeicher I_j , der eine Teilmenge der im gesamten Netz verfügbaren Information enthält. Da ein Sender nicht immer die gesamte ihm bekannte Information senden kann bzw. möchte, wird die Auswahlfunktion $\psi: \mathcal{P}(I) \rightarrow \mathcal{P}(I)$ als Ausgangsfilter verwendet. Die Änderung des lokalen Speichers durch die von der Kommunikation unabhängige Programmierung des Kommunikationsteilnehmers wird durch das Funktional $\tau: \mathcal{P}(I) \rightarrow \mathcal{P}(I)$ repräsentiert. Anschaulich bedeutet diese Modellierung, dass ein Knoten sich seine eigenen Informationen durch τ modifiziert zusendet. Die Funktion $\phi: \mathcal{P}(I) \rightarrow \mathcal{P}(I)$ modelliert die Tatsache, dass ein Teilnehmer nicht zwangsläufig die gesamte empfangene Information speichern muss bzw. kann, sondern nur einen Teil davon. Eine schematische Darstellung findet man in Abbildung 1.

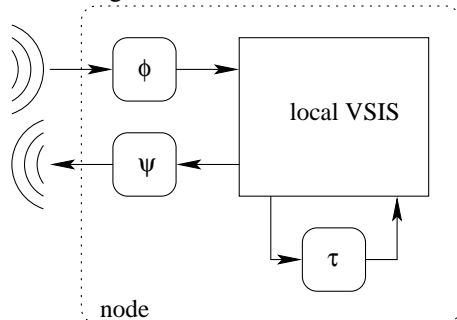


Abbildung 1 Lokaler VSIS mit angeschlossenen Filterfunktionen

2.2 Virtual Shared Information Spaces auf Basis von XCast

Die XCast-Kommunikationsteilnehmer wählen über die Funktionen Φ und Ψ Informationen aus und verteilen diese dann innerhalb der Nachbarschaften. Die Funktion τ verändert in jedem Zeitschritt den lokalen Speicher eines Kommunikationspartners.

Das Systemverhalten lässt sich mit Hilfe einer rekursiven Funktion beschreiben, die einen Zustand des Systems in einen Folgezustand überführt:

$$\text{XCast}_{\phi, \psi, \tau}^t(I_1, \dots, I_n) := \text{BCast}(\text{XCast}_{\phi, \psi, \tau}^{t-1}(I_1, \dots, I_n))$$

wobei BCast die IEEE 802.11 Broadcast-Kommunikation abbildet. Da bei der datenzentrierten Kommunikation die Entscheidung darüber, welche Information gesendet bzw. empfangen wird, durch die Funktionen Φ und Ψ getroffen wird, kann man auf die Nutzung von Netzwerkadressen zur Auswahl der Empfänger verzichten. Dies ist insbesondere bei Funk auch eine dem Medium angepasste Vorgehensweise.

Betrachtet man die Kommunikation über einen längeren Zeitraum, wird schließlich die gesamte verfügbare Information soweit wie möglich im Netzwerk verteilt sein. Im Idealfall, d.h. bei einem zusammenhängenden bzw. vollständig vermaschten Netz, wird jeder lokale Speicher die für den jeweiligen Knoten vollständige

Information beinhalten. Formal lässt sich dieser Zustand als Fixpunkt (vgl. z.B. [9]) der XCast Funktion beschreiben. Der VSIS, welcher mit Hilfe der XCast-Kommunikationstechnik etabliert wird, ist dann definiert als der kleinste Fixpunkt der XCast Funktion.

Der Informationsgehalt des VSIS wird durch die drei Auswahlfunktionen Ψ , Φ und τ sowie durch die Nachbarschaftsrelation bestimmt. Da man die Nachbarschaftsrelation a priori nicht festlegen kann und diese in Folge von Mobilität ständigen Änderungen unterworfen ist, bleiben einem zur Definition des VSIS nur die Auswahlfunktionen Ψ , Φ und τ , deren Design deshalb möglichst robust gegenüber dem Einfluss der Nachbarschaftsrelation sein sollte. Einige Designstrategien werden im nächsten Abschnitt beschrieben, nachdem auf die Informationskodierung eingegangen wurde.

3 XML-basierte Datenrepräsentation

Damit Dokumente bzw. die darin kodierten Informationen verarbeitet werden können, muss die Informationsinklusionsrelation \subset und die Informationsvereinigungsoperation \sqcup definiert werden. Die Informationsinklusionsrelation gibt an, ob eine Information I_1 in einer anderen Information I_2 enthalten ist. Formal ist \subset als Halbordnung über der Menge aller Informationen definiert. Die Informationsvereinigungsoperation vereinigt zwei Informationen I_1 und I_2 zu einer neuen Information $I_1 \sqcup I_2$, so dass $I_1 \subset I_1 \sqcup I_2$ und $I_2 \subset I_1 \sqcup I_2$ ist. Falls die Informationen nicht vereinigt werden können, liefert die Vereinigungsoperation \sqcup einen Fehler, d.h. $I_1 \sqcup I_2 = \perp$.

Der Vorteil einer XML-basierten Datenrepräsentation liegt in der leichteren Verarbeitung semistrukturierter, hierarchischer Daten. Im Folgenden werden wir zeigen, warum dazu eine Informationsrepräsentation auf der Basis von Markup-Sprachen wie XML geeignet ist. Im weiteren Text wird immer angenommen, dass eine Information I in einem XML Dokument kodiert ist. Die Begriffe Information und Dokument werden deshalb gleichbedeutend verwendet. Wir sprechen von einem atomaren Dokument, falls dieses nur einen Eintrag enthält. Ein Dokument kann wiederum in einem Paket gekapselt versendet werden.

Betrachtet man beispielsweise die beiden folgenden XML-Dokumente, so ist das obere Dokument ein Teildokument des unteren Dokumentes:

```
<AddressList>
  <Entry>
    <Name>Reuter</Name>
    <Mail>flr@informatik.uni-kiel.de</Mail>
  </Entry>
</AddressList>


---


<AddressList>
  <Entry>
    <Name>Reuter</Name>
    <Mail>flr@informatik.uni-kiel.de</Mail>
    <Phone>+49 431 8807296</Phone>
  </Entry>
  <Entry>
    <Name>Luttenberger</Name>
    <Mail>nl@informatik.uni-kiel.de</Mail>
  </Entry>
</AddressList>
```

Die Vereinigung der beiden Dokumente liefert somit das untere Dokument. Formal könnte man die Informationsinklusionsrelation \subset für Adresslisten wie folgt definieren:

1. Eine Adressliste L_1 ist in einer Adressliste L_2 enthalten, wenn für alle Einträge E_1 der Adressliste L_1 gilt, dass es einen Eintrag E_2 in der Adressliste L_2 gibt, so dass E_1 eine Teilinformation von E_2 ist.
2. Ein Eintrag E_1 ist Teilinformation eines Eintrages E_2 genau dann, wenn die Namen der beiden Einträge übereinstimmen und wenn in Eintrag E_1 eine Mailadresse (bzw. Telefonnummer) angegeben ist, dann muss diese mit der Mailadresse (bzw. Telefonnummer) des Eintrages E_2 übereinstimmen.

Die Informationsvereinigungsoperation \sqcup für Adresslisten kann durch folgenden Algorithmus implementiert werden:

1. Vereinige die Einträge der beiden Adresslisten zu einer neuen Adressliste, indem die Listen konkateniert werden.
2. Reduziere die neu entstandene Eintragsliste, indem jeder Eintrag E_1 entfernt wird, zu dem es einen Eintrag E_2 (mit $E_1 \neq E_2$) in der Liste gibt, so dass E_1 eine Teilinformation von E_2 ist.

Zusammen mit der Möglichkeit von XML, Elementen Attribute zuzuweisen, kann man die obigen Definitionen noch verfeinern, indem man z.B. auf Datums- und Zeitattribute an den Adresseinträgen reagiert. So könnte z.B. erreicht werden, dass veraltete Information bei der Vereinigungsoperation entfernt wird.

4 Strategien zur Definition der Auswahlfunktionen Φ , Ψ und τ

Obwohl das Konstrukt des VSIS neu ist, haben sich bereits einige Strategien abgezeichnet, die bei der Definition der Auswahlfunktionen hilfreich sind. Die wichtigsten Strategien bzgl. der verschiedenen Auswahlfunktionen werden nun kurz beschrieben.

Die Funktion Φ bestimmt, welche Information weiterverarbeitet wird. Damit ist Φ auch dafür verantwortlich, dass die Informationsvereinigung keinen Fehler liefert, d.h. nur Information in den Verarbeitungsprozess aufgenommen werden, die auch verarbeitet werden kann. Für eine erfolgreiche Informationsverarbeitung ist es daher wichtig, den Empfangsfilter Φ möglichst tolerant zu machen, um die Menge an verarbeitbarer Information möglichst groß zu halten. Dieses Vorgehen findet man auch schon bei der Internetarchitektur [7], wo von einem Empfänger eine tolerante Auslegung des Kommunikationsprotokolls verlangt wird. Um diesen Gedanken auf die Informationsfilterung bei XML-kodierten Daten übertragen zu können, war es nötig, die Semantik von XML Schema [16, 5] zu erweitern. Vorarbeiten der Autoren findet man in [15].

Die Funktion Ψ bestimmt das Sendeverhalten. Es wurde bereits erwähnt, dass das Ziel bei der Definition der Filterfunktionen sein muss, möglichst robust gegenüber der Mobilität der Knoten zu sein. Dieses muss vor allem beim Design des Sendeverhaltens berücksichtigt werden. Es kann sich nicht darauf verlassen werden, dass eine Information, welche gesendet wird, auch tatsächlich beim Empfänger ankommt. Es hat sich gezeigt, dass es

günstig ist, der Funktion Ψ einen probabilistischen Anteil zu geben, so dass Information zu randomisierten Zeitpunkten gesendet oder auch erneut gesendet wird.

Die Funktion τ ist für Transformation der Information im lokalen Speicher verantwortlich. Zum einen ermöglicht sie die Implementierung der von der Kommunikation unabhängigen Funktionalität, zum anderen kann durch geeignete Wahl von τ Meta-Information nachgeführt werden. Empfängt ein Knoten beispielsweise Zeitinformationen von anderen Knoten, so können diese mit Hilfe der τ Funktion mit der lokalen Uhr fortgeführt, d.h. approximiert werden.

5 Ein VSIS zur Verteilung endlicher Informationen

In diesem Abschnitt wird durch geeignete Wahl der Funktionen Φ , Ψ und τ ein VSIS etabliert, welcher die Verteilung einer endlichen Informationsmenge innerhalb eines Netzwerkes vornimmt.

5.1 Definition von Φ und Ψ

Das einfachste bekannte Verfahren zum Fluten eines Netzes mit Information ist das Simple Flooding. Es kodiert die zu verbreitende Information in einem Paket, welches durch eine Knoten- und Paketidentifikation eindeutig gemacht wird. Empfängt ein Knoten ein solches Paket, leitet er es nur weiter, wenn er die Knoten- und Paketidentifikation noch nicht gesendet hat. Durch dieses Verfahren wird sichergestellt, dass ein Paket von jedem Knoten genau einmal weitergeleitet wird. Kritisch ist die Größe der Liste, welche die bereits gesendeten Identifikatoren enthält. Ist die Liste zu klein, so werden Pakete unter Umständen mehrmals weitergeleitet und ist die Liste zu groß, so können ressourcenbeschränkte Knoten diese eventuell nicht mehr speichern.

Aufbauend auf dem Simple Flooding Verfahren könnte man die Informationsverteilung dadurch erreichen, dass jeder Knoten periodisch seine initiale Information I_j^0 an alle Nachbarn versendet. Genügt die Senderate der Mobilität des Netzes, so wird die gesamte im Netz verfügbare Information an jeden Knoten verteilt. Dieses Verfahren erzeugt einen beträchtlichen Kommunikations-Overhead, da Pakete, die dieselbe Information tragen, aufgrund unterschiedlicher Knoten- und Paketidentifikationen separat weitergeleitet werden.

Durch eine geeignete Wahl der XCast Filterfunktionen Φ , Ψ und τ erhält man eine inhaltsbasierte Version des Simple Flooding, welche im folgenden mit *Content-based Simple Flooding* (CBSF) bezeichnet ist. Das CBSF trifft die Entscheidung, ob ein Paket weitergeleitet werden soll, nicht mehr anhand von Identifikatoren, sondern anhand der Information selbst.

Die Funktionen $\Phi_{CBSF}(I)$ und $\Psi_{CBSF}(I)$ nutzen einen Sendepuffer *send_buffer*, welcher initial leer ist und als Warteschlange für zu versendende Daten dient. Der Empfangsfilter $\Phi_{CBSF}(I)$ verwirft jede Information, die dem Knoten j bereits lokal bekannt ist, d.h. für welche $I \subset I_j$ gilt. Ist die Information lokal nicht bekannt, so wird diese ungefiltert weitergeleitet und zum erneuten Senden dem Sendepuffer übergeben.

```

 $\Phi_{CBSF}(I) := \text{if } I \subset I_j \text{ then}$ 
    return  $\emptyset$ ;
else
     $send\_buffer := send\_buffer \sqcup I$ ;
    return  $I$ ;
fi

```

Die Sendefilterfunktion Ψ_{CBSF} gibt den Sendepuffer und zusätzlich mit der Wahrscheinlichkeit w_s die initiale Information I_j^0 des Knoten j zurück. Somit ist sichergestellt, dass neue Information im Netzwerk verbreitet wird und die initiale Information des Knotens j trotz evtl. Paketverluste von anderen Knoten empfangen werden kann.

```

 $\Psi_{CBSF}(I) := local\_buffer := send\_buffer$ ;
     $send\_buffer := \emptyset$ ;
     $local\_buffer := local\_buffer \sqcup \begin{cases} I_j^0 & \text{W'keit } w_s \\ \emptyset & \text{W'keit } 1-w_s \end{cases}$ 
return  $local\_buffer$ ;

```

Eine lokale Informationsverarbeitung findet nicht statt, d.h. die Funktion τ wird als $\tau(I) := I$ definiert. Der kleinste Fixpunkt bezeichnet somit den gewünschten Zustand, in dem die gesamte im Netzwerk verfügbare Information — sofern möglich — allen Knoten bekannt ist.

5.2 Ein Beispielszenario

Den *ns2*-Simulationen zur Verteilung endlicher Information liegt folgende Applikationsidee zugrunde: Auf einer Fläche von 2500 Quadratmetern befinden sich 50 Teilnehmer einer Veranstaltung. Jeder Teilnehmer besitzt ein elektronisches Adressbuch, das mit IEEE 801.11b Hardware ausgestattet ist. Der Idee des CBSF folgend wird mittels XCast bei jedem Teilnehmer eine Teilnehmerliste der Veranstaltung generiert.

Der initiale Speicher I_j^0 des Knotens j besteht aus einer Adressliste mit genau einem Eintrag, welcher den Besitzer des elektronischen Adressbuch repräsentiert. Ein Eintrag besteht aus einem 30 Zeichen langen Namen und einer 30 Zeichen langen Mailadresse. Die Teildokumentenbeziehung und die Dokumentenvereinigung sind analog der Definitionen im Abschnitt 3. Das Funktional τ verändert die Information nicht, d.h. $\tau(I) = I$.

Die Simulationen wurden bewusst mit stationären Knoten gemacht, denn Mobilität kann die Informationsverteilung – wie z.B. in [4] und [17] gezeigt wird – unterstützen.

Die 500 Simulationsläufe ergaben, dass in über 99% der Fälle mit dem gewählten Sendejitter 0.8 bis 1.2s und der Sendewahrscheinlichkeit $w_s = 0.1$ nach spätestens 60s bei jedem Teilnehmer eine komplette Teilnehmerliste gebildet wurde.

5.3 Analytische Bewertung

Die Anzahl der benötigten Pakete, um den VSIS mit Hilfe des vorgestellten Modells zu etablieren, entspricht maximal der Anzahl der von allen Filterfunktionen im Netzwerk generierten atomaren Dokumenten. Durch Aggregation von Teildokumenten und aufgrund der initialen Verteilung und Eigenschaften der Information im Netzwerk kann die Anzahl der Pakete deutlich reduziert werden. Die gleichmäßige Verteilung der Information in unserem Beispiel ist dabei der ungünstigste Fall.

Wir bezeichnen mit $p_\psi(t)$ und $p_\phi(t)$ die maximale Anzahl der von Ψ_{CBSF} bzw. Φ_{CBSF} pro Zeitintervall erzeugten Pakete. Es gilt also

$$\sum_{t=0}^T (p_\psi(t) + p_\phi(t)).$$

In jedem Zeitintervall wird von Ψ_{CBSF} mit der Wahrscheinlichkeit w_s ein neues atomares Dokument erzeugt. Somit ergibt w_s multipliziert mit der Knotenzahl n die maximal von der Funktion Ψ_{CBSF} erzeugten Pakete im gesamten Netz:

$$p_\psi(t) \leq n \cdot w_s(t) = n \cdot w_s$$

Die Zahl der von der Funktion Φ_{CBSF} pro Zeitintervall erzeugten Pakete lässt sich abschätzen, indem man die Wahrscheinlichkeit $w_r(t)$ einführt, mit welcher ein Knoten im Zeitintervall ein atomares Dokument weiterleitet. Demnach erhält man:

$$p_\phi(t) \leq n \cdot w_r(t)$$

Zusammenfassend kann man nun die Anzahl der erzeugten Pakete abschätzen, indem man p_ψ und p_ϕ einsetzt:

$$\sum_{t=0}^T (p_\psi(t) + p_\phi(t)) \leq T \cdot n \cdot w_s + n \cdot \sum_{t=0}^T w_r(t)$$

In diesem speziellen Beispiel kann man für die Anzahl der weitergeleiteten atomaren Dokumente annehmen, dass jeder Knoten jedes Dokument genau einmal weiterleitet, d.h. es gilt $n \cdot \sum_{r=0}^T w_r(t) = n^2$.

Ein allgemeingültigerer und damit weiterführender inhaltsbasierter Ansatz zur Abschätzung der Anzahl der übertragenen Pakete basiert auf der Betrachtung der Wahrscheinlichkeit w_α neue Information zu empfangen.

Aus der Definition der Eingangsfilterfunktion Φ_{CBSF} ergibt sich, dass diese Wahrscheinlichkeit der Wahrscheinlichkeit entspricht, dass eine Information weitergeleitet wird. Sei $docs(t)$ die Anzahl der bis zum Zeitpunkt t von einem Knoten empfangenen atomaren Dokumente. Weiterhin nehme man an, dass atomare Dokumente sequentiell empfangen werden, d.h. indiziert werden können. Man erhält so die Äquivalenz

$$w_r(t) = w_\alpha(docs(t)).$$

Dies entspricht einer Achsentransformation (Zeit nach Anzahl empfangener Dokumente). Es folgt

$$\sum_{t=0}^T w_r(t) = \sum_{t=0}^T w_\alpha(docs(t)) = \sum_{d=1}^{docs(T)} w_\alpha(d).$$

Die Wahrscheinlichkeit $w_\alpha(d)$, dass ein Knoten nach d erhaltenen atomaren Dokumenten ein ihm unbekanntes Dokument erhält, geht mit wachsendem d gegen Null, da die Menge der Information im Netzwerk endlich ist. Im allgemeinen Fall ist die Wahrscheinlichkeit, dass ein Knoten j die Information I als d -tes Dokument empfängt, z.B. durch eine Wahrscheinlichkeitsdichte $\chi(I, d, j)$ gegeben. Im Sinne der Vereinfachung nehmen wir an, dass diese Wahrscheinlichkeit gleichverteilt sei – was im betrachteten Beispiel der Fall ist.

Man betrachte die Folge von d atomaren Dokumenten, die ein Knoten empfangen hat, wobei das d -te Element das als letztes empfangene atomare Dokument ist. Dieses darf nicht im Präfix der Folge enthalten sein, damit es sich um ein neues atomares Dokument handelt. Für den Postfix kann aus n Informationen jeweils mit der Wahrscheinlichkeit $\frac{1}{n}$ eine ausgewählt werden. Die Wahr-

scheinlichkeit, ein vom Postfix unterschiedliches atomares Dokument zu erhalten, ist gleich $\frac{n-1}{n}$. Für das Präfix der Länge $d-1$ ergibt sich also $\left(\frac{n-1}{n}\right)^{d-1}$. Damit kann man folgende Gleichung herleiten:

$$\begin{aligned} \sum_{d=1}^{docs(T)} w_{\varphi}(d) &= \sum_{d=1}^{docs(T)} \underbrace{\left(\frac{n-1}{n}\right)^{d-1}}_{\text{Präfix}} \cdot \underbrace{\frac{1}{n} \cdot n}_{\text{Postfix}} \\ &= \frac{n}{n-1} \sum_{d=1}^{docs(T)} \left(\frac{n-1}{n}\right)^d \\ &= \frac{n}{n-1} \sum_{d=1}^{docs(T)} \frac{1}{k^d} \quad \text{mit } \frac{1}{k} = \frac{n-1}{n} < 1 \end{aligned}$$

Die Summe konvergiert, da es sich um eine geometrische Reihe handelt und $\frac{n-1}{n} < 1$ ist. Für den Grenzwert D lässt sich zeigen, dass

$$\text{für } T \rightarrow \infty: \sum_{d=1}^{docs(T)} w_{\varphi}(d) = n = D = 50$$

gilt, was mit der vorherigen Betrachtung übereinstimmt (siehe auch Abbildung 2).

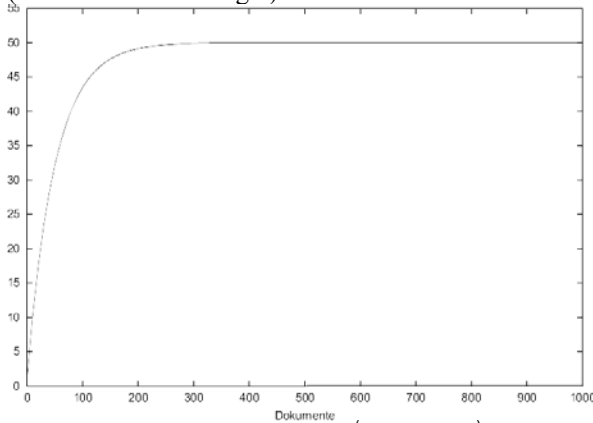


Abbildung 2: Darstellung von $\left(\sum_{d=1}^{D'} w_{\varphi}(d)\right)(D')$ über der Anzahl empfangener Dokumente D' .

Geht man allgemein davon aus, dass man diesen Dokumentengrenzwert D bestimmen kann, lässt sich die Gesamtzahl der erzeugten Pakete in Abhängigkeit von der Gesamtzeit T , der Knotenzahl n und der Sendewahrscheinlichkeit w_s durch

$$\text{Anzahl erzeugter Pakete} \leq T \cdot n \cdot w_s + n \cdot D$$

abschätzen. Diese Abschätzung ist aus zweierlei Gründen interessant: Erstens hängt die Zahl der weitergeleiteten Pakete nicht mehr von der Zeit ab und zweitens wird eine direkte Beziehung zwischen der Informationsrepräsentation und der Zahl der Pakete hergestellt. Aufgrund dieser linearen Abhängigkeit der Paketanzahl von dem Dokumentengrenzwert D kann die Skalierung von Algorithmen gezielt durch den Einsatz einer effizienten Kodierung verbessert werden.

5.4 Simulationsergebnisse

In Abbildung 3 ist die Anzahl der generierten Pakete in den Simulationen über die Zeit dargestellt. Vom Empfangsfilter Φ wurden in der Simulation im Schnitt 2313 Pakete erzeugt. Dies ist etwas weniger als die berechnete Anzahl von $n \cdot D = n^2 = 2500$ und kann durch die durchschnittlich 100 Kollisionen auf der MAC-Schicht erklärt werden.

Der Sendefilter hat in den Simulationen durchschnittlich 751 Pakete erzeugt, was ziemlich genau mit dem erwarteten Wert von $T \cdot n \cdot w_s = 150 \cdot 50 \cdot 0.1 = 750$ übereinstimmt. Die Zahl der tatsächlich auf der MAC-Schicht übertragenen Pakete ist noch kleiner als die Anzahl der generierten Pakete. Dies hängt damit zusammen, dass durch die Vereinigungsoperationen in den Filterfunktionen mehrere Informationen in ein MAC-Paket kodiert werden (vgl. Abbildung 4). Dies spiegelt sich auch in der Größe der Pakete wider (vgl. Abbildung 5). Wie in [13] gezeigt wird, wirkt sich die Übertragung größerer Pakete in funkbasierten Netzen positiv aus.

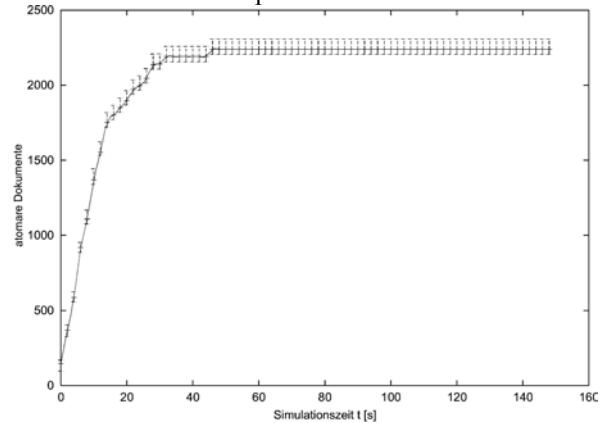


Abbildung 3 Anzahl der von Empfangsfilter Φ insgesamt generierten atomaren Dokumente

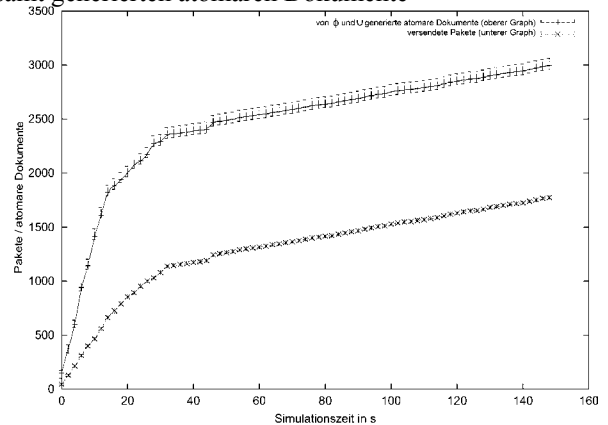


Abbildung 4 Generierte atomare Dokumente und gesendete Pakete.

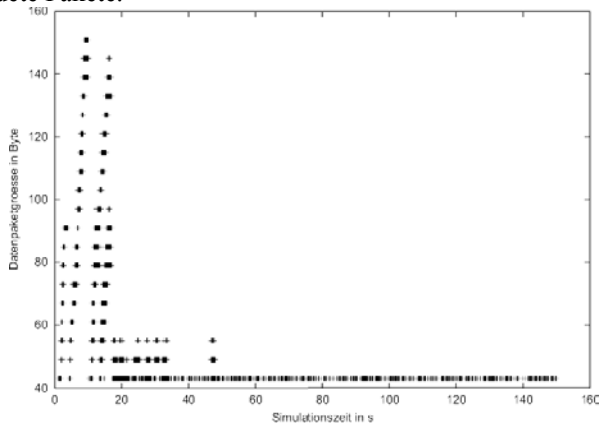


Abbildung 5: Größe der Pakete auf der MAC-Schicht aufgetragen über die Zeit.

6 Verwandte Arbeiten

Ein bekanntes inhaltsbasiertes Paradigma aus dem Bereich der Sensornetzwerke zur Koordination verteilter Mess-, Kommunikations- und Berechnungsapplikationen

ist "Directed Diffusion"[12]. Es ist datenzentriert in dem Sinne, dass (Name, Value)-Paare übertragen werden und jeder Knoten des Netzes der Applikation gewahr sein muss. Das Verfahren ist Grundlage und Stellvertreter vieler anderer Forschungsarbeiten. Keine dieser Arbeiten wendet sich jedoch detaillierter dem Problem der Informationsrepräsentation und deren Abhängigkeit in der Leistungsbewertung zu.

In [10] werden inhaltsbasierte Protokolle vorgestellt, welche versuchen, durch Betrachtung des Inhalts von Paketen, den Kommunikations-Overhead zu verringern. Diese zeichnen sich vor allem dadurch aus, dass die Idee des Application Level Framing[8] umgesetzt wird. Zudem werden Meta-Daten eingesetzt, um den Umfang der zu übertragenden Anwendungsdaten zu beeinflussen, was die Anzahl der generierten Pakete reduziert.

Einen Überblick über effiziente jedoch nicht inhaltsbasierte Verfahren zum Verteilen von Informationen durch Fluten von Netzen findet man in [18].

Das aus dem Bereich der verteilten Datenbanken stammende Verfahren des Rumor Spreading[20] sorgt durch ein dem Verhalten von Epidemien und Gerüchten nachempfundenes Verfahren für die Synchronisation verteilter Datenbestände. Diese Verfahren nutzen jedoch keine Broadcast-Kommunikation; vielmehr ist ein wesentlicher Bestandteil dieser Verfahren, den nächsten Knoten auszuwählen, welcher "infiziert" werden soll.

Arbeiten und Anwendungen aus dem Bereich der XML-basierten Middleware wie UPnP[2] und Web-Service[3] oder auch XML-basierte Tupel Spaces[6, 19,14] basieren meist auf dem klassischen Client/Server Modell und sind daher für Mobile Ad-Hoc Netze nicht ohne weiteres geeignet. Ihre Schwäche besteht darin, dass durch den Ausfall eines Knotens (des Servers) das gesamte System zusammenbricht.

7 Zusammenfassung und Ausblick

Im ersten Teil dieser Arbeit wurde gezeigt, wie ein datenzentriertes, XML-basiertes Kooperationsmodell aussehen könnte und welche Freiheitsgrade dessen Programmierung bietet. Das vorgestellte Konzept des *virtual shared information space* zeichnet sich durch seine flexible Programmierung anhand der Filterfunktionen, seine XML-basierte Datenzentriertheit und seine für Mobile Ad-Hoc Netze geeignete serverlose Architektur aus. Im zweiten Teil dieses Artikels wurde mit Hilfe des Kooperationsmodells des *virtual shared information space* ein Dienst implementiert, der die Verteilung endlicher Datenmengen innerhalb von Mobilien Ad-Hoc Netzen vornimmt. Sowohl die berechnete als auch die simulierte Performance Analyse ergab, dass die Anzahl der benötigten Pakete unter anderem linear von einem "Dokumentengrenzwert" D abhängt. Mit Hilfe dieser Beziehung können nun gezielt leistungsfähigere Algorithmen für Mobile Ad-Hoc Netze entwickelt werden, in deren Design der Dokumentengrenzwert D berücksichtigt wird. Zukünftige Arbeiten untersuchen den Zusammenhang zwischen der XML-Kodierung und diesem Grenzwert. Diese Arbeit ist im Rahmen des von der DFG geförderten SWARMS Projekts [1] entstanden, welches in Zusammenarbeit mit der TU-Braunschweig bearbeitet wird.

Literatur

1. SWARMS - Software Architecture for Radio-based Mobile Systems. <http://www.swarms.de>.
2. UPnP Forum. <http://www.upnp.org/>.
3. Web services. <http://www.w3.org/2002/ws/>.
4. A. Beaufour, M. Leopold, and P. Bonnet. Smart-tag based data dissemination. In First ACM International Workshop on Wireless Sensor Networks and Applications (*WSNA*), pages 68–77, September 2002.
5. P. V. Biron and A. Malhotra. XML Schema Part 2: Datatypes. W3C Recommendation, May 2001. <http://www.w3.org/TR/xmlschema-2/>.
6. G. Cabri, L. Leonardi, and F. Zambonelli. XML Dataspaces for the Coordination of Internet Agents. *Applied Artificial Intelligence*, 15(1):35–58, January 2001.
7. B. Carpenter. RFC 1958: Architectural principles of the Internet, June 1996. Status: INFORMATIONAL.
8. D. D. Clark and D. L. Tennenhouse. Architectural considerations for a new generation of protocols. In Proceedings of the ACM symposium on Communications architectures & protocols, pages 200–208. ACM Press, 1990.
9. G. Goos. Vorlesungen über Informatik Band 1: Grundlagen und funktionales Programmieren. Springer, 2 edition, 1997.
10. W. R. Heinzelmann, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Seattle, Washington, USA, August 1999. ACM.
11. A. L. Hors, P. L. H'egaret, L. Wood, G. Nicol, J. Robie, M. Champion, and S. Byrne. Document Object Model (DOM) Level 3 Core Specification. W3C Working Draft, October 2002. <http://www.w3.org/TR/2002/WD-DOM-Level-3-Core-20021022/>.
12. C. Intanagonwivat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proceedings of the sixth annual international conference on Mobile computing and networking*, pages 56–67. ACM Press, 2000.
13. D. A. Maltz. *On-Demand Routing Multi-hop Wireless Mobile Ad Hoc Networks*. PhD thesis, Carnegie Mellon University, 2001.
14. D. Moffat. XML-Tuples and XML-Spaces, March 1999. <http://uncled.oit.unc.edu/XML/XMLSpaces.html>.
15. F. Reuter and N. Luttenberger. Cardinality Constraint Automata: A core technology for efficient XML Schema-aware parsers. <http://www.swarms.de/publications/cca.pdf>.
16. H. S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn. XML Schema Part 1: Structures. W3C Recommendation, May 2001. <http://www.w3.org/TR/xmlschema-1/>.
17. A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, Duke University, Durham, North Carolina, July 2000. Available at <ftp://ftp.cs.duke.edu/dist/techreport/2000/2000-06.ps>.
18. B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *The Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Lausanne, Switzerland, June 2002. ACM.
19. P. Wycko.. T Spaces. *IBM Systems Journal*, 37(3), 1998. <http://www.research.ibm.com/journal/sj/373/wycko..html>
20. A. Demers, D. Green, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturings, D. Swineheart and D. Terry. Epidemic Algorithms for Replicated Database Maintenance. In *Proceedings of the Sixth Symposium on Principles of Distributed Computing*, pages 1-12, August 1987.