

Stabiles Routing von Endsystem-Multicast-Strömen in Mobilen Ad-hoc-Netzen

Peter Baumung
Institut für Telematik
Universität Karlsruhe (TH)
baumung@tm.uka.de

Abstract: Endsystem-basierte Protokolle sind viel versprechende Ansätze zur Realisierung von Gruppenkommunikationsdiensten in mobilen Ad-hoc-Netzen. Ein hierfür aufgebautes Overlay-Netz ist aufgrund der möglichen Mobilität der Teilnehmer stetig an die Topologie des zugrunde liegenden Netzes anzupassen. Für Anwendungen, welche auf kontinuierlichen Datenströmen basieren, können sich hieraus jedoch Probleme ergeben, da kurzzeitig auftretende Inkonsistenzen des Overlay-Routings oftmals zu inakzeptablen Latenzzeiten und zahlreichen Paketverlusten bei der Zustellung der Daten führen. Dieser Beitrag stellt ein einfaches Konzept vor, wie diese Probleme mit geringem zusätzlichen Aufwand vermieden werden können.

1 Einführung und Motivation

Mobile Ad-hoc-Netze (kurz auch MANETs genannt) bieten die Möglichkeit, ohne Einsatz jeglicher fest installierter Infrastruktur zu kommunizieren. Auch im Hinblick auf Gruppenkommunikation und hier insbesondere im Rahmen von eLearning-Anwendungen, bieten MANETs einen interessanten Ansatz zur Erbringung diverser Dienste. Denkbar ist zum Beispiel die Übertragung von Whiteboard-Strömen während einer Vorlesung oder die Verteilung relevanter Unterlagen an eine Gruppe von Studierenden. Aufgrund starker Ähnlichkeiten zwischen MANETs und der allgemeinen Struktur gängiger Peer-to-Peer-Systeme, bietet sich die Realisierung der oben genannten Gruppenkommunikationsdienste anhand von Endsystem-basierten Protokollen an. Diese bauen ein so genanntes Overlay-Netzwerk auf, welches die Mitglieder einer Gruppe anhand von Transportverbindungen miteinander verknüpft. Der von einer Anwendung ausgehende Datenstrom kann so gemäß eines Overlay-Routing-Protokolls über die Transportverbindungen des Overlays zu den Gruppenmitgliedern geleitet werden [LTM99, GM03, BI04, BZ04].

Im Gegensatz zu im Festnetz betriebenen Endsystem-basierten Anwendungen ist in MANETs eine kontinuierliche Anpassung des aufgebauten Overlay-Netzes erforderlich. Dies ergibt sich aus der Tatsache, dass die in MANETs vorhandene Knotenmobilität sich in Form von schwankenden Qualitäten der Transportverbindungen auswirkt. Insbesondere ist bei der Auseinanderbewegung zweier miteinander verbundenen Gruppenmitglieder in der Regel mit einer Verschlechterung der Verbindungsqualität beider Knoten zu rechnen,

da eine zunehmende Anzahl von Übertragungsabschnitten überbrückt werden muss. Lässt man die aufgrund der Knotenmobilität verursachten Veränderungen der zugrunde liegenden Netztopologie außer Acht, wirkt sich dies negativ auf die Effizienz eines Overlay-Netzes und somit auch auf die Qualität des darüber erbrachten Dienstes aus.

Um eine möglichst hohe Dienstgüte zu bewahren, ist eine kontinuierliche Überwachung und Bewertung der Transportverbindungen durchzuführen. Schlechte Verbindungen sind abzubauen und gegebenenfalls durch bessere zu ersetzen. Die Topologie eines Overlay-Netzes beschreibt im Laufe des Kommunikationsvorhabens demnach eine gewisse Evolution. Diese kann jedoch bei der Verteilung eines kontinuierlichen Datenstroms entlang des Overlays zu Problemen führen. Grund hierfür ist die langsame Propagation der Overlay-Routing-Information in MANETs. Letztere wird in den meisten Fällen periodisch zwischen den Overlay-Knoten ausgetauscht, wobei die Periodizität derart gewählt wird, um einen akzeptablen Kompromiss zwischen der Aktualität der Routing-Information und der dafür benötigten Netzbelastung zu bilden. Wegen der in MANETs häufig auftretenden Paketverluste ist es somit möglich, dass relevante Änderungen der Overlay-Topologie, welche eine Anpassung des Overlay-Routings erfordern, für längere Zeit (bis zu mehreren 10 Sekunden) verzögert werden. In diesem Zeitraum ist das innerhalb des Overlay-Netzes stattfindende Routing inkonsistent, da die Weiterleitung der Daten die veränderte Overlay-Topologie noch nicht berücksichtigt. Unter diesen Umständen ist es möglich, dass der Datenstrom manchen Gruppenmitgliedern aufgrund soeben abgebauter Transportverbindungen auf unbestimmte Zeit nicht zugestellt werden kann. Hieraus resultieren oftmals zahlreiche Paketverluste, welche aufgrund begrenzter Paketpuffer und abhängig von der Dauer der Inkonsistenz nicht behoben werden können.

Dieser Beitrag ist wie folgt gegliedert. Abschnitt 2 stellt einen generischen Mechanismus vor, mit Hilfe dessen das für den Datenstrom eingesetzte Routing auch im Falle einer sich ändernden Overlay-Topologie stabil bleibt. In Abschnitt 3 wird anschließend knapp ein Protokoll vorgestellt, welches zur Evaluation des vorgestellten Verfahrens eingesetzt wurde. Ein direkter Vergleich von Simulationsergebnissen, wie sie mit und ohne Verwendung des vorgestellten Verfahrens erzielt wurden, ist Abschnitt 4 zu entnehmen. Abschnitt 5 bietet abschließend eine Zusammenfassung der Arbeit sowie einen kurzen Ausblick.

2 Entkopplung von Datenfluss und Overlay-Netz

Das einführend beschriebene Problem zahlreicher Paketverluste aufgrund inkonsistenter Routing-Information kann umgangen werden, indem die Verteilung des eigentlichen Datenstroms von dem dafür eingesetzten Overlay-Netz entkoppelt wird. Letzteres wird anschließend nicht mehr direkt zur Verteilung der Daten eingesetzt, sondern primär dafür, um Gruppenmitglieder zu informieren, von welchen weiteren Gruppenmitgliedern Daten einer Multicast-Quelle effizient bezogen werden können.

Hierfür versendet eine Multicast-Quelle periodisch zusätzliche Kontrollpakete des Typs `UpdatePath`. Diese werden entlang des Overlays gemäß des eigentlichen Overlay-Routing-Protokolls zu den Empfängern geleitet. Letztere erfahren hierbei von denjenigen Overlay-

Knoten, welche auf dem Datenpfad einer bestimmten Quelle direkt übergeordnet sind. Aufgrund des Overlay-Routing-Protokolls stellen diese Elternknoten auch jeweils die Knoten dar, von welchen Daten einer Quelle am effizientesten bezogen werden können.

Ein Empfänger sendet mit einer gewissen Periodizität Pakete des Typs `RegisterAsChild` an seinen derzeitigen Elternknoten, um diesem den Wunsch zu signalisieren, Daten einer Multicast-Quelle zu beziehen. Ein jeder Overlay-Knoten führt demnach unabhängig vom eigentlichen Overlay-Routing-Protokoll eine zweite Forwarding-Tabelle, in welcher diejenigen Knoten vermerkt werden, von denen derzeit `RegisterAsChild`-Pakete empfangen werden. Overlay-Knoten greifen nun zur Weiterleitung von Multicast-Daten auf diese Forwarding-Tabelle zurück, und nicht mehr auf das Overlay-Routing-Protokoll.

Tritt zu einem Zeitpunkt eine Änderung der Overlay-Topologie ein, äußert sich die Inkonsistenz des Overlay-Routings für einen betroffenen Empfänger lediglich durch das Ausbleiben der `UpdatePath`-Pakete von Seiten des Elternknotens. Unabhängig davon verharret der Empfänger jedoch bei seinem zuletzt bekannten Elternknoten, an welchen auch weiterhin `RegisterAsChild`-Pakete gerichtet werden. Der Datenempfang bleibt somit stabil. Nach Auflösung der Inkonsistenz werden `UpdatePath`-Pakete unter Umständen von einem anderen Knoten empfangen. Da dieser gemäß des Overlay-Routing-Protokolls nun einem günstigeren Elternknoten entspricht, findet ein Wechsel des Elternknotens statt. Der Datenbezug über den zuvorigen Elternknoten bleibt jedoch bestehen, bis der erfolgreiche Bezug durch den neuen Elternknoten anhand des Empfangs des ersten Datenpakets von diesem erkannt wurde. Es ist somit möglich, kurzzeitig identische Datenpakete von zwei unterschiedlichen Elternknoten zu beziehen.

Eine deutliche Reduktion des Aufwands kann erreicht werden, indem Pakete des Typs `UpdatePath` nur im Falle einer sich ändernden Overlay-Topologie versendet werden. Werden beispielsweise von einem Knoten bereits `RegisterAsChild`-Pakete empfangen, so sind an diesen Knoten keine `UpdatePath`-Pakete zu senden, da dieser bereits Kenntnis seines optimalen Elternknotens hat. Prinzipiell können zur Verteilung der Kontrollpakete sowie des eigentlichen Datenstroms die gleichen Transportverbindungen genutzt werden. Jedoch sind Transportverbindungen, die auf Kontrollebene nicht mehr benötigt werden, erst dann abzubauen, wenn diese auch auf Datenebene keine Verwendung mehr finden.

3 Anwendung auf das NICE-MAN-Protokoll

Das in Abschnitt 2 beschriebene Verfahren wurde für das NICE-MAN-Protokoll [BI04] implementiert. Bei diesem handelt es sich um eine Adaption des hierarchischen NICE-Protokolls [BBK02] aus dem Festnetzbereich, welche darauf zielt, den Eigenschaften von MANETs (Knotenmobilität und Broadcast-Medium) Rechnung zu tragen. Um die allgemeine Zustellrate des Protokolls zu erhöhen, wurden zusätzlich die in [BZK04] beschriebenen elementaren Zuverlässigkeitsmechanismen implementiert. Diese beinhalten Sendewiederholungen anhand von `NACK`-Paketen sowie eine einfache, auf Fenstertechnik basierende Fluss- und Staukontrolle. Letztere setzt `ACK`-Pakete ein, um das Sendefenster einer Multicast-Quelle freizugeben. Hierzu gibt jeder Overlay-Knoten eine bestimmte Se-

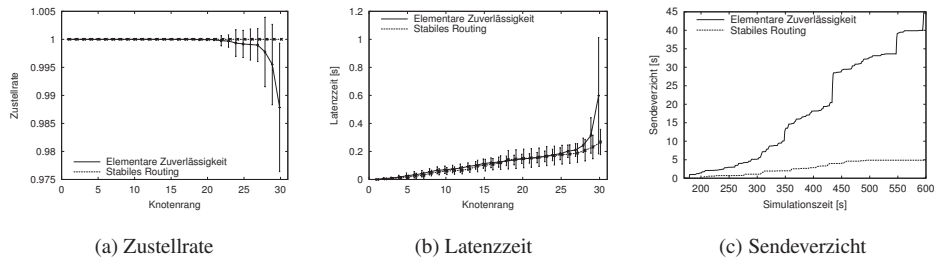


Abbildung 1: Simulationsergebnisse

quenznummer an seinen auf dem Datenpfad einer Quelle übergeordneten Knoten weiter. Die Sequenznummer entspricht dabei dem Minimum aller (von untergeordneten Knoten) empfangenen und der eigenen, höchsten Reihenfolge-getreuen Sequenznummer. Auf diese Weise wird eine Multicast-Quelle indirekt über den Empfangsstatus aller Empfänger informiert, wobei das Problem der Senderimplosion umgangen wird. Ist das Sendefenster der Quelle erschöpft, blockiert diese, so dass das Medium ausschließlich zur Behebung von Paketverlusten verwendet wird. Von Interesse ist bei diesem Verfahren also auch die Dauer, für welche eine Quelle aufgrund eines erschöpften Sendefensters blockiert ist.

Um den zur Entkopplung des Datenflusses benötigten Aufwand zu minimieren, wurden die in Abschnitt 2 entworfenen `RegisterAsChild`-Pakete in Form der ohnehin periodisch versendeten und für die Fluss- bzw. Staukontrolle benötigten `ACK`-Pakete realisiert.

4 Simulationsergebnisse

Eine Evaluierung des aus Abschnitt 3 resultierenden Protokolls wurde im Rahmen von Forschungsarbeiten zur Untersuchung von Diensten für mobile Lerngruppen erhalten. Hierzu wurden mit Hilfe von *QualNet* Simulationen durchgeführt, in welchen sich auf einer Fläche von $1000m \times 1000m$ eine Gesamtheit von 80 Knoten mit einer Sendereichweite von $150m$ und einer Medienbandbreite von $2 \frac{Mbit}{s}$ aufhält. Die Multicast-Gruppe besteht aus 30 Knoten, welche einer Verbands-basierten Bewegung mit einer Höchstgeschwindigkeit von $1 \frac{m}{s}$ unterliegen. Die übrigen 50 Knoten bewegen sich nach zufälligem Muster mit maximal $2 \frac{m}{s}$. Die Multicast-Quelle sendet über 5 Minuten hinweg einen Datenstrom, welcher aus 4 Paketen pro Sekunde mit jeweils 512 Bytes besteht. Zur Glättung der Messungen wurden die Ergebnisse 20 unterschiedlicher Szenarien gemittelt. Zwei der Diagramme stellen Knotenränge dar. Diese entstehen, indem innerhalb jedes Szenarios die Ergebnisse eines Knotens zunächst gemittelt und diese dann sortiert aufgetragen werden. Zur besseren Unterscheidung der Kurven wurden diese leicht entlang der x-Achse versetzt.

Abbildung 1 ist eine deutliche Verbesserung der Leistungen des Protokolls zu entnehmen. Während einige der ursprünglich im Laufe von Routing-Inkonsistenzen entstandenen Paketverluste nicht behoben werden können, wird nach der Entkopplung des Datenflusses

eine ideale Zustellrate für alle Knoten erzielt. Ebenso behält die Kurve der Latenzzeiten nun auch bei den schlechtesten Knoten ihren annähernd linearen Charakter bei. Eine deutliche Verbesserung ist auch für die Dauer zu verzeichnen, für welche die Multicast-Quelle blockiert ist bzw. auf das Versenden weiterer Daten verzichtet. Grund hierfür ist, dass Empfänger selbst im Falle von Inkonsistenzen des Overlay-Routings in der Lage sind, Daten einer Multicast-Quelle zu beziehen. Sie fallen deshalb bei der Verteilung des Datenstroms nicht zurück, so dass sie nach Auflösung der Inkonsistenzen kaum Paketverluste aufweisen und somit auch keine lang andauernde Blockierung des Senders verursachen.

5 Zusammenfassung und Ausblick

In diesem Beitrag wurde ein generischer Mechanismus beschrieben, mit Hilfe dessen Routing-Inkonsistenzen bei Endsystem-basierten Multicast-Streaming-Anwendungen in MANETs mit geringem zusätzlichen Aufwand vermieden werden können. Die Effektivität des Verfahrens wurde demonstriert, indem ein entsprechend angepasstes Protokoll simulativ evaluiert wurde. In Kombination mit elementaren Zuverlässigkeitsmechanismen wie Übertragungswiederholungen sowie einer Flusskontrolle konnte dabei eine ideale Zustellrate an eine Gruppe von 30 mobilen Empfängern erreicht werden.

Weitere Arbeiten auf diesem Gebiet werden sich damit beschäftigen, Overlay-Protokolle im Hinblick auf eine Entkopplung des Datenflusses effizienter zu gestalten. Hierdurch werden Reduktionen des zur Pflege des Overlay-Netzes anfallenden Kontrollflusses erhofft, welche sich auch durch eine bedeutend geringere Medienbelastung äußern können.

Literatur

- [BBK02] Banerjee, S., Bhattacharjee, B., und Kommareddy, C.: Scalable Application Layer Multicast. In: *ACM SIGCOMM 2002*. Pittsburgh, PA, USA. June 2002.
- [BI04] Bloedt, S.: Efficient End System Multicast for Mobile Ad Hoc Networks. In: *The First International Workshop on Mobile Peer-to-Peer Computing*. Orlando, Florida, USA. March 2004.
- [BZ04] Baumung, P. und Zitterbart, M.: Effiziente Endsystem-basierte Multicast-Dienste in mobilen Ad-hoc Netzen. In: *Lecture Notes in Informatics*. 18. DFN-Arbeitstagung, Düsseldorf, Germany. June 2004.
- [BZK04] Baumung, P., Zitterbart, M., und Kutzner, K. Improving Delivery Ratios for Application Layer Multicast in Mobile Ad-hoc Networks. Accepted for the 4th Workshop on Applications and Services in Wireless Networks, Boston, Massachusetts, USA. 2004.
- [GM03] Gui, C. und Mohapatra, P.: Efficient Overlay Multicast for Mobile Ad hoc Networks. In: *The Wireless Communications and Networking Conference (WCNC)*. New Orleans, Louisiana, USA. March 2003.
- [LTM99] Liu, M., Talpade, R. R., und McAuley, A.: AMRoute: Adhoc Multicast Routing Protocol. Technical Report TR 99-1. Center for Satellite and Hybrid Communication Networks. 1999.