

# Effizientes Peer-to-Peer-Distributionssystem für multimediale Inhalte

Thorsten Strufe

Technische Universität Ilmenau

Institut für Praktische Informatik und Medieninformatik

thorsten.strufe@tu-ilmenau.de

**Abstract** Im vorliegenden Papier wird ein Distributionssystem für multimediale Inhalte auf Basis des Peer-to-Peer-Paradigmas vorgeschlagen, welches bekannte Nachteile dieser Technologie durch die Einführung von relativer Lokalität und einer losen Struktur umgeht. So wird der große Kommunikationsaufwand, der häufig in solchen Systemen auftritt, minimiert. Zusätzlich skaliert das System auch bei einer großen Anzahl Teilnehmer gut, ermöglicht das Publizieren von Inhalten an beliebigen Knoten, kommt ohne breitbandig angebundene Kommunikationspartner aus und bietet sichere und zuverlässige Kommunikation.

## 1 Einleitung

Der Begriff Peer-to-Peer unterliegt einer Inflation, wie sie bei interessanten Technologien häufig auftritt. In diesem Papier wird daher von der ursprünglichen Bedeutung ausgegangen: es sollen darunter dezentrale Systeme verstanden werden, die aus lose gekoppelten Knoten mit gleichem Funktionsumfang bestehen. Daraus folgt, dass alle Knoten eines Peer-to-Peer-Systems die gleichen Funktionen erfüllen oder zumindest erfüllen können. Es handelt sich auf der Organisationsebene bei dem Peer-to-Peer-Modell damit um das Gegenteil des Client-Server-Modells, da es in einzelnen Ansätzen lose strukturierte, grundsätzlich jedoch dezentrale Systeme umfasst.

Bei multimedialen Inhalten handelt es sich in aller Regel um Videos oder Animationen, welche sich als zu übertragende Daten insbesondere durch ihre Größe, sowie besonders hohe Anforderungen an eine rechtzeitige Übertragung auszeichnen.

Die Verteilung multimedialer Daten in verteilten Organisationen ist bis heute problematisch. Bestehende Lösungsansätze, wie in der Vergangenheit häufig genutzte zentralisierte Ansätze, sind für diese Zwecke nicht immer ausreichend. Sie führen bei der Verteilung multimedialer Inhalte zu einem serverseitig sehr hohen Kommunikationsaufwand und bedürfen einer zusätzlichen Einführung von Mechanismen zur Lastverteilung. Darüber hinaus bildet der Server einen Single-Point-of-Failure, welcher durch Ausfall das ganze System zum Erliegen bringt.

Ein weiteres in diesem Zusammenhang auftretendes Problem ist der Mangel an Push-Lösungen, die es ermöglichen Daten sicher von einem zentralen Punkt aus an die angebotenen Kommunikationspartner zu verteilen, ohne dass diese in den Vorgang eingreifen müssen.

Aus diesen Gründen wird ein Distributionssystem vorgeschlagen, welches den Anforderungen dieses Anwendungsfeldes durch die Einführung einer dezentralen Peer-to-Peer-Semantik entspricht.

## **2 Umfeld und Anforderungen**

Das vorliegende Anwendungsfeld zeichnet sich durch eine äußerst hohe Dynamik und unzuverlässigen Aufbau aus. Viele Knoten sind nicht durchgängig erreichbar, da sie nur eingeschränkte Betriebszeiten haben oder ihnen lediglich schmalbandige und unzuverlässige Einwahlverbindungen zur Verfügung stehen. So führen die Einschränkungen zu einem Netz aus häufig ausfallenden und sich erneut registrierenden, sehr lose gekoppelten Knoten.

Um ein für das gegebene Anwendungsfeld sinnvoll einsetzbares System zu erhalten müssen die Nachteile bestehender Technologien wie Gnutella [6], insbesondere die sehr hohe Netzlast und die mangelnde Lastverteilung vermieden werden. Daraus resultierend ergeben sich die Anforderungen nach effizientem und ressourcenschonendem Datentransfer, einer auch bei grosser Nutzeranzahl und grossen Datenmengen gewährleisteten Skalierbarkeit, sicherer und zuverlässiger Kommunikation und der Beständigkeit des entstehenden Netzwerks, trotz der Unzuverlässigkeit einzelner Knoten. Die Wartezeit bis zur vollständigen Übermittlung einer Datei soll möglichst gering und die Übertragung zuverlässig sein.

## **3 Vergleichbare Ansätze**

Es gibt bereits eine große Anzahl bestehender Peer-to-Peer-Systeme, die jedoch durchgängig eine Reihe von Unzulänglichkeiten aufweisen. Sie bestehen gewöhnlich aus einer Menge zufällig verbundener Knoten denen eine begrenzte Liste anderer Knoten bekannt ist [6]. Zur Lokalisierung von Ressourcen flutet ein suchender Knoten das Netz mit einer Anfrage. Solche Suchvorgänge werden zumeist durch einen Ereignishorizont, welcher etwa durch einfache Time-To-Live-Felder realisiert wird, ein geschränkt. Führt eine Suche zu keinem Ergebnis, so wird sie mit einem erweiterten Suchhorizont wiederholt und das Netz erneut geflutet.

Erhält der suchende Knoten eine Trefferantwort von einem beliebigen Knoten irgendwo innerhalb des Suchhorizonts, so baut er mit diesem eine Verbindung auf, beginnt mit dem Kopiervorgang der gesamten Datei und ignoriert dabei weitere einkommende Treffer.

Die eingesetzte Lokalisierungsmethodik ist nicht zuverlässig, erzeugt eine hohe Netzlast und skaliert sehr schlecht. In der Literatur finden sich diverse Ansätze, diese Eigenschaften zu verbessern. Einzelne Lösungsansätze [1, 2, 3] versuchen das Manko durch eine Strukturierung des Netzes zu umgehen. Diese führen jedoch zu Einschränkungen der Suchbarkeit von Ressourcen und mangelnder Toleranz gegenüber einer unregelmäßigen Betriebszeit oder dem Ausfall von Knoten.

Ein weiterer Nachteil für die Übertragung grosser Datenmengen ist der Zugriff auf lediglich eine einzelne Quelle und das Ignorieren weiterer Quellen. Hat der Quellknoten keinen breitbandigen Netzzugang, so kann es zu sehr langen Übertragungszeiten kommen. Fällt er unter Umständen während der Übertragung aus oder wird die Verbindung

unterbrochen, so muss der gesamte Vorgang erneut begonnen werden. Auch für diese Problematik gibt es Lösungsansätze, etwa durch sinnvolles Setzen von Checkpoints oder die Unterteilung der Daten in Fragmente [4]. Allerdings bieten diese nur eingeschränkte Suchmöglichkeiten und skalieren schlecht.

Schließlich existiert die Problematik der Netzpartitionierung: durch den Ausfall eines oder mehrerer Knoten kann es zur Fragmentierung des Peer-to-Peer-Netzes in mehrere, nicht miteinander verbundene Teilnetze kommen.

## **4 Modell des Systems**

Bei dem vorgeschlagenen System handelt es sich um eine Peer-to-Peer-Topologie, die lose strukturiert wird.

Um den Kommunikationsaufwand einzuschränken, wird zum einen eine relative Lokalität berücksichtigt und zum anderen für Knoten die Rollen der Nodes und Supernodes [5] eingeführt. Supernodes übernehmen hierbei die Funktionalität von Namens-, Lokalisierungs- und Benachrichtigungsdiensten.

Alle Knoten sind nach der Initialisierung zunächst normale Nodes, können bei Bedarf jedoch zu Supernodes rekonfiguriert werden. Näheres zur Rekonfiguration ist in Punkt 7.2 beschrieben.

### **4.1 Lokalität**

Lokalität ist als absolute geographische Größe ohne Zugriff auf die Router oder ein GPS sehr schwer zu erlangen. Aus diesem Grund wird eine relative Lokalität eingeführt, welche aus einer Kombination aus IP-Hops und gemittelten Antwortzeiten besteht.

Jeder Knoten bestimmt die Entfernungen zu den ihm bekannten anderen Knoten und legt danach die für ihn lokal und die entfernt erscheinenden Knoten fest.

Diese Vorgehensweise verhindert das einzelne Nodes mit einer schlechten und unzuverlässigen Anbindung an das Netz von allen anderen Nodes als entfernt eingestuft werden und so aus dem Netz herausfallen.

### **4.2 Nodes**

Nodes halten eine Liste mit ihnen bekannten anderen Nodes, die Peerliste (auch Host-Cache[6], Warehouse[7] etc.) vor, die lokale und entfernte Supernodes, sowie die Nodes mit welchen zuletzt kommuniziert wurde, inklusive der jeweiligen relativen Entfernung enthält.

Ein Node meldet sich bei den für ihn lokal erscheinenden Supernodes an und übergibt dabei eine Liste seiner Ressourcen. Kommt es infolge des Datenaustausches zu einer Verbindung zwischen entfernten Nodes, die nach den Kriterien eines der beiden Nodes lokal sind, so tauschen sie die Informationen über ihre Supernodes aus.

Verdrängt in der Peerliste eines Nodes ein neuer oder bislang entfernter Supernode einen lokalen, so meldet sich der Node zunächst zusätzlich bei ihm an, mittelt die jeweiligen Entfernungen über einen begrenzten Zeitraum und meldet sich danach von dem als entfernt ermittelten Supernode ab.

Das Speichern entfernter Knoten führt zur Resistenz des Systems gegenüber einer Fragmentierung durch den Ausfall einzelner Knoten.

### **4.3 Supernodes**

Für die globale Signalisierung zur Suche und zur Benachrichtigung übernehmen die Supernodes die Funktion eines Gateways ins Netz: Lokalisierungsnachrichten Push-Informationen werden vollständig über die Supernodes geleitet und von ihnen lokal an ihre Nodes weitergegeben.

Sie bieten zusätzlich zur Nodefunktionalität also Namens-, Lokalisierungs- und Benachrichtigungsdienste an.

Zu diesem Zweck unterhalten sie umfangreichere Peerlisten, in welchen zusätzlich zu ID, Adresse und Entfernung der einzelnen Knoten Informationen über ihre Ressourcen abgelegt sind. Die Peerliste unterteilt sich in Supernodes und lokale Nodes sowie entfernte Nodes, welche durch LRU-Strategie in der Liste verdrängt werden.

## **5 Information**

Für jede zum Austausch bestimmte Datei wird zum Zeitpunkt der Publizierung mittels einer Hashfunktion ein Bezeichner generiert. Das Hashen sichert die eindeutige Zuordnung auch bei mehrfachem Auftreten gleicher Namen und ermöglicht zusätzlich eine Versionierung der Daten.

Durch eine anschließende Fragmentierung und die automatische Replikation der Daten an den jeweils anfragenden Nodes ergibt sich eine Lastverteilung von selbst, da vollständig kopierte Fragmente für andere suchende Nodes bereitgestellt werden.

## **6 Kommunikation**

Die Kommunikation setzt sich aus Daten- und Kontrollfluß, respektive Transfer und Signalisierung zusammen. Der Transfer findet zweigeteilt statt: Um eine gute Verteilung zu erreichen werden Kopiervorgänge von lokal nicht vorhandenen Fragmenten mit einer hohen Priorität belegt. Dabei werden zuerst zu den in kürzester Entfernung liegenden Knoten Verbindungen aufgebaut. Gleichzeitig werden regelmäßig am Supernode die eigenen Ressourcen aktualisiert und die Liste der lokal vorhandenen Fragmente abgeglichen. Kommt es bei einer Transferverbindung zu niedrigen Übertragungsraten, so wird versucht das Fragment parallel von anderen Knoten zu kopieren.

Für den Transfer werden TCP-Verbindungen genutzt. Nach erfolgtem Verbindungsaufbau werden Informationen über Ressourcen und unter Umständen Supernodes ausgetauscht, bevor der Kopiervorgang beginnt.

Die Signalisierung zum Zweck weiterer Dienste wird nur über die Supernodes realisiert. Such- und Pushanfragen sowie Rekonfigurationsdaten richten die Nodes an ihre Supernodes, welche diese an ihre lokalen Nodes und an die anderen Supernodes verteilen. Um im Supernode-Netz Broadcasts zu vermeiden werden Message Chains [7] (welche in jüngeren Veröffentlichungen, leicht abgewandelt, auch als „Random Walker“ [8] auftauchen) genutzt.

## 7 Funktionalitäten

Neben der Distribution beliebiger Daten, der zentralen Aufgabe des Systems, sind einige weitere Funktionalitäten nötig. Hierzu gehören die Lokalisierung von Ressourcen und die Rekonfiguration der Knoten zu Supernodes oder Nodes.

### 7.1 Suche, Distribution und Abruf

Um die Möglichkeit zu haben, eine Ressource aus dem Netz zu kopieren, muss ein Knoten sie zunächst auffinden.

Grundsätzlich muss dabei zwischen zwei Funktionen unterschieden werden: der Suche nach einem Namen oder einer Beschreibung und der Lokalisierung einer dadurch gekennzeichneten Ressource bzw. deren Replika.

Die Suche einer Ressource wird durch einfaches Substring-Matching realisiert. Wird an einem Node die Suche nach einer Ressource ausgelöst, so stellt er eine Suchanfrage an einen seiner lokalen Supernodes. Supernodes senden eingehende Suchanfragen an die anderen Supernodes weiter und durchsuchen die lokal gemeldeten Ressourcen nach dem gesuchten String. Bei Treffern, werden Namen, Beschreibung und Hashwert der gefundenen Ressource an den anfragenden Knoten zurückgeschickt.

Um eine Ressource zu Lokalisieren wird eine Suchanfrage nach ihrem Hashwert gestellt, die von Supernodes bei Treffern mit der Adresse des Quellknoten beantwortet wird.

Die Distribution von Ressourcen verläuft in mehreren Schritten:

- 1) **Einstellung:** Der einstellende Node meldet die Ressource (Namen und Hashwert) dem lokalen Supernode.
- 2) **Push (Supernodes)** Verteilung des Pushsignals an alle Supernodes.
- 3) **Push (Nodes):** Empfangende Supernodes informieren lokale Nodes und legen eine Liste über Quellen und lokale Fragmente an.
- 4) **Lokalisierung:** Informierte Knoten lokalisieren Quellen und ermitteln Informationen über lokale oder gerade kopierte Fragmente
- 5) **Kopieren** Knoten bauen Verbindungen zu Quellen auf und kopieren Fragmente.

Während des gesamten Verlaufs werden die Listen der Supernodes über Quellknoten sowie vorhandene und gerade im Kopiervorgang befindliche Fragmente aktualisiert.

Soll eine Ressource erneut ohne ein Pushsignal abgerufen werden, wird anstelle der Push-Vorgänge die oben beschriebene Suche eingesetzt.

### 7.2 Rekonfiguration

Jeder Knoten wird manuell mit einer Peerliste, welche bestehende Nodes enthält, konfiguriert und als einfacher Node initialisiert. Die Erweiterung eines Nodes zu einem Supernode kann zwei Ursachen haben.

Zum einen versucht ein Node immer mit einer zu ermittelnden Anzahl lokaler Supernodes verbunden zu sein und rekonfiguriert sich selbst zu einem Supernode, falls er diese nicht finden kann. Zum anderen kann es vorkommen, dass ein Supernode „überläuft“, also nicht mehr in der Lage ist, alle bei ihm als lokal angemeldeten Nodes zu bedienen. In diesem Fall sucht er in seiner Peerliste eine wiederum zu

determinierende Anzahl leistungsfähiger Nodes und informiert sie darüber, dass sie Supernodes werden sollen.

Die Anzahl lokaler Supernodes und zu informierender Nodes ergibt sich aus der gemittelten Zuverlässigkeit des Netzes. Die Leistungsfähigkeit, nach welcher zu rekonfigurierende Nodes ausgesucht werden, ergibt sich aus ihren statistischen Betriebszeiten und Bandbreiten.

Die Rück-Konfiguration eines Supernodes erfolgt, wenn die Anzahl der lokalen Supernodes größer als die Anzahl lokaler Nodes ist, unter der Voraussetzung, dass die Mindestzahl lokaler Supernodes dadurch nicht unterschritten wird.

## 8 Zusammenfassung und Ausblick

Durch das vorgeschlagene System kann die Verteilung multimedialer Inhalte deutlich vereinfacht werden. Trotz des Einsatzes von Peer-to-Peer-Technologien zur besseren Lastverteilung kommt es nicht zu Systemausfällen durch die typischen Probleme der Netzwerküberlastung und der Fragmentierung entstehender Netze.

Für zukünftige Betrachtungen bleiben jedoch folgende Forschungsaufgaben offen:

- Bessere Konzepte für die Lokalität
- Einführung sicherer Gruppen innerhalb unsicherer Netze
- Verbesserung der Signalisierung, insbesondere der Lokalisation
- Untersuchung unterschiedlicher Replikationsmechanismen
- Ermittlung optimaler Fragmentierung der Daten und
- Leistungsfähiger Cachingmechanismen.

## 9 Referenzen

- [1] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek and H. Balakrishnan: *Chord: A scalable peer-to-peer lookup service for Internet applications*. Technical Report TR-819, MIT, March 2001.
- [2] A. Rowstron and P. Druschel: *Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems*. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), November, 2001.
- [3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker: *A Scalable Content-Addressable Network*. SIGCOMM01, August 27-31, 2001, San Diego, CA, USA
- [4] I. Clarke: *A Distributed Decentralised Information Storage and Retrieval System*. Unpublished undergraduate thesis, University of Edinburgh. 1999. <http://freenet.sourceforge.net/>
- [5] *FastTrack*; <http://www.fasttrack.nu/>
- [6] *The Gnutella Protocol Specification V. 0.4*; <http://rfc-gnutella.sourceforge.net/>
- [7] M. Wulff, P. Kropf, and H. Unger: *Message Chains and Disjunct Path for Increasing Communication Performance in Large Networks*. In P. Kropf et al., editor, *Distributed Communities on the Web 2000*.
- [8] Qin Lv, Pei Cao, Edith Cohen, Kai Li, Scott Shenker: *Search and Replication in Unstructured Peer-to-Peer Networks*. In Proceedings of 16th ACM International Conference on Supercomputing (ICS'02), New York, USA, June 2002