

# Peer-to-Peer-Systeme und -Anwendungen

Ralf Steinmetz<sup>1</sup>, Klaus Wehrle<sup>2</sup>, Oliver Heckmann<sup>1</sup>

<sup>1</sup>Multimedia Kommunikation, TU Darmstadt

<sup>2</sup>Protocol-Engineering & Verteilte Systeme, Universität Tübingen

<sup>1</sup>{Ralf.Steinmetz, Oliver.Heckmann}@KOM.tu-darmstadt.de

<sup>2</sup>Klaus.Wehrle@uni-tuebingen.de

## 1 Einführung

Unter dem Begriff „*Peer-to-Peer*“ etabliert sich derzeit ein neues und höchst interessantes Paradigma für die Kommunikation im Internet. Obwohl ursprünglich nur für die sehr pragmatischen und rechtlich umstrittenen Dateitauschbörsen entworfen, können die Peer-to-Peer-Mechanismen zur verteilten Nutzung unterschiedlichster Betriebsmittel genutzt werden und neue Möglichkeiten für internetbasierte Anwendungen eröffnen.

Zahlreiche Telekommunikationsunternehmen berichten, dass zur Zeit das Verkehrsaufkommen von Peer-to-Peer-Diensten deutlich mehr als 50% des Gesamtvolumens ausmacht, in Spitzenzeiten sogar mehr als 75%. Aufgrund des weiterhin anhaltenden Wachstums des Internets hinsichtlich Teilnehmerzahlen und Datenaufkommen, aber auch durch die gestiegenen Anforderungen des ständig wachsenden Anwendungsspektrums, lassen sich zahlreiche Anwendungen mit den traditionellen, auf Client-Server-Ansätzen basierenden Methoden oft nur noch mit erheblichem Aufwand realisieren. Es lassen sich in diesem Zusammenhang drei grundlegende Anforderungen an Internet-basierte Anwendungen im zukünftigen Internet identifizieren:

- Die *Skalierbarkeit* ist eine elementare Voraussetzung, um den weiterhin steigenden Teilnehmerzahlen sowie dem immensen Ressourcenbedarf (Bandbreite, Speicherplatz, Rechenkapazität) bestimmter Anwendungen gerecht zu werden. So gilt es von Beginn an Flaschenhälse zu vermeiden, um auch ein Wachstum um mehrere Größenordnungen effizient bewältigen zu können.
- *Sicherheit* und *Zuverlässigkeit* spielen im Zuge einer wachsenden Anzahl gezielter Angriffe (sog. „Distributed Denial of Service“-Attacken) auf zentrale Dienste eine entscheidende Rolle für die Verfügbarkeit strategisch wichtiger und sicherheitsrelevanter Dienste. Auch eine zensurresistente Speicherung von Daten und Anonymität gewinnen durch aktuelle Entwicklungen entscheidend an Bedeutung.
- Mehr *Flexibilität* und insbesondere *Dienstgüte* sind für die einfache und schnelle Integration neuer Dienste ein entscheidendes Erfolgskriterium künftiger Internet-Technologien. So konnten dringend benötigte Dienste, wie etwa Gruppenkommunikation oder Mobilität, bis heute nicht mit einer adäquaten Qualität realisiert werden.

Es wird zunehmend deutlich, dass die seit Anfang der achtziger Jahre zunehmend eingesetzten Client-Server-basierten Anwendungen den neuen Anforderungen des Internets nicht mehr umfassend gerecht werden. Insbesondere zentrale Instanzen deren Ressourcen zum Flaschenhals werden können, die gezielte Angriffspunkte darstellen oder, die aufgrund ihrer strategischen Lage im Netzwerk nicht ohne hohe Kosten modifiziert werden können - bereiten zunehmend Probleme. An dieser Stelle versprechen nun die Konzepte des Peer-to-Peer-Networking bzw. Peer-to-Peer-Computing<sup>1</sup> durch einen grundlegenden Paradigmenwechsel die genannten Probleme einfacher lösen zu können.

## 2 Definition

Ausgehend von einer ersten Definition in [Ora01] und deren Präzisierung in [SW04] wird unter einem Peer-to-Peer-System ein „*sich selbst organisierendes System gleichberechtigter, autonomer Einheiten (Peers)*“ verstanden, „*das vorzugsweise ohne Nutzung zentraler Dienste auf der Basis eines Rechnernetzes mit dem Ziel der gegenseitigen Nutzung von Ressourcen operiert*“ – kurzum ein *System mit vollständig dezentraler Selbstorganisation und Ressourcennutzung*. Neben diesen grundlegenden Prinzipien lassen sich Peer-to-Peer-Systeme nach unserem Verständnis durch die folgenden Charakteristika beschreiben, wobei oftmals nur im Idealfall alle Eigenschaften gleichzeitig zutreffen:

### **Zur dezentralen Ressourcennutzung:**

1. Die relevanten Betriebsmittel (Bandbreite, Speicherplatz, Rechenkapazität) werden möglichst gleichmäßig verteilt genutzt und befinden sich an den sog. „Kanten“ der Rechnernetze, bei den „Peers“. Damit können aus Sicht des Netzwerkes Peer-to-Peer-Systeme als die konsequente Weiterentwicklung des Ende-zu-Ende-Prinzips [SRC84] aufgefasst werden, das als eines der entscheidenden Prinzipien für den bisherigen Erfolg des Internets gilt.
2. Eine Menge von Peers nutzen untereinander die von ihnen wiederum zur Verfügung gestellten Betriebsmittel. Die bekanntesten Beispiele solcher Betriebsmittel sind Speicherplatz (in Form von Dateien, u.a. mit Musik, Video und Programmen als Inhalt) und Prozessorleistung. Daneben sind aber auch Konnektivität, menschliche Präsenz oder aber geographische Nähe zu nennen. Beispiele hierfür sind Instant Messaging und Gruppenkommunikationsanwendungen.
3. Peers sind über das Netz miteinander verbunden und meist weltweit verteilt.
4. Ein Peer ist oftmals nicht stets unter derselben Internet-Adresse zu identifizieren (variable Konnektivität). Typischerweise erhalten Peers beim Verbinden an das Netz jeweils eine neue Internet-Adresse dynamisch zugeteilt. Peers sind auch häufig über

---

<sup>1</sup>Es sei angemerkt, dass wir hier Peer-to-Peer-Networking nicht von Peer-to-Peer-Computing unterscheiden wollen, sondern uns auf „Peer-to-Peer“ als Eigenschaft, Charakter, Verfahren bzw. Mechanismus konzentrieren.

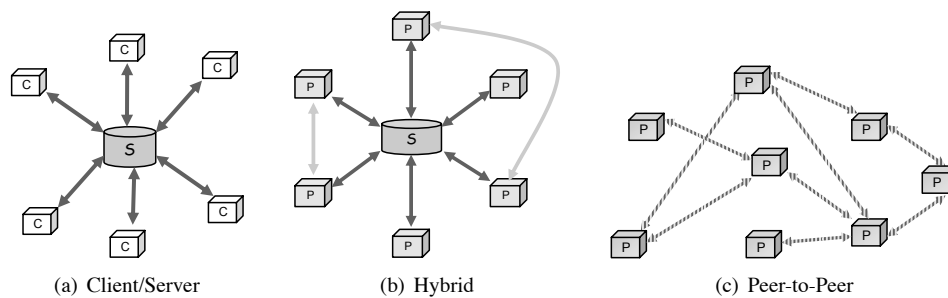


Abbildung 1: Klassifikation von Peer-to-Peer-Systemen

längere Zeit nicht am Netz angeschlossen oder gar abgeschaltet. Aufgrund dieser Unwägbarkeiten aber auch aus zahlreichen weiteren Gründen führen daher viele Peer-to-Peer-Anwendungen oberhalb der normalen Internet-Adressierung individuelle Adressräume ein. So werden Inhalte meist durch unstrukturierte Identifikatoren adressiert, die mit Hilfe von Hash-Funktionen aus den jeweiligen Inhalten gewonnen werden. Dies hat zur Folge, dass Daten nicht mehr mit ihrer Lokation (Adresse des Servers) identifiziert werden, sondern vielmehr durch den Inhalt selbst. Befinden sich beispielsweise mehrere Instanzen eines Datums im System, so kann jede dieser Instanzen das Ziel der Wegewahl werden. Das Auffinden eines Datums in einem Peer-to-Peer-System erfolgt somit durch eine inhaltsbasierte Wegewahl (content based routing) an Stelle der ortsbezogenen Wegewahl im Internet.

#### Zur dezentralen Selbstorganisation:

5. Zur eigentlichen Nutzung der Betriebsmittel interagieren Peers direkt miteinander. Dies erfolgt im Allgemeinen ohne jegliche zentrale Steuerung. Hierin drückt sich wiederum ein ganz zentrales Merkmal von Peer-to-Peer-Systemen aus, das sie von traditionellen Client-Server-Systemen unterscheidet: während letztere Koordination ausgehend vom Server als ordnendes Paradigma verwenden, bauen Peer-to-Peer-Systeme auf Kooperation zwischen gleichberechtigten Partnern. Der Hintergrund dieses Wechsels begründet sich vor allem in der Vermeidung von Flaschenhälsen und dem Verlust von Zuverlässigkeit im Vergleich zu zentralisierten Lösungen.
6. Der Zugriff auf bzw. Transfer der gemeinsam benutzten Betriebsmittel erfolgt direkt zwischen den Peers. Hierfür existiert kein zentraler Dienst. Peer-to-Peer-Systeme stehen für eine grundsätzliche Dezentralisierung der Steuerung, auch wenn im Sinne der Leistungssteigerung dann auch in existierenden Systemen unter Umständen zentrale Elemente eingefügt werden, beispielsweise um die Suche nach Betriebsmitteln effizienter zu gestalten. Diese Systeme auch oft als hybride Peer-to-Peer-Systeme bezeichnet(vgl. Abb. 1b).

7. Peers selbst besitzen sowohl Client als auch Server-Funktionalität, d.h. Peers sind beides zugleich (siehe Abb. 1c). Dies unterscheidet Peer-to-Peer-Systeme wesentlich von herkömmlichen Systemen mit asymmetrischer Funktionalitätsverteilung (siehe Abb. 1a) und führt sowohl zu erhöhter Flexibilität hinsichtlich der bereitgestellten Funktionalität als auch zu sehr unterschiedlichen Anforderungen an den Entwurf von Peer-to-Peer-Systemen.
8. Peers sind untereinander gleichberechtigt. Sie besitzen individuell die vollständige Autonomie bezüglich ihrer jeweiligen Betriebsmittel.
9. Das Auffinden von Ressourcen sollte möglichst ohne jeglichen zentralen Dienst erfolgen (in Abb. 1a und 1b sind zentrale Dienste dafür vorhanden, in Abb. 1c nicht). Idealerweise geschieht die gesamte Steuerung selbstorganisierend bzw. im ad-hoc Modus. Wie bereits erwähnt kann hiervon im Sinne der Leistungssteigerung zwar abgewichen werden, ohne jedoch den dezentralen Charakter von Peer-to-Peer-Systemen völlig zu verletzen. Dies führt zu Peer-to-Peer-Systemen mit hybrider Struktur (vgl. Abb. 1b).

## 2.1 Paradigmenwechsel

Peer-to-Peer ist damit keinesfalls nur ein Verfahren zum Austausch von Dateien, sondern vielmehr ein grundlegendes Entwurfsprinzip für verteilte Systeme, in dem deutlich der Paradigmenwechsel von *Koordination zu Kooperation*, von *Zentralisierung zu Dezentralisierung* und von *Kontrolle hin zu Anreizen* erkennbar ist. Eine Vielzahl wichtiger Fragestellungen stehen im Zusammenhang mit geeigneten Anreizsystemen. Eine faire Balance im „Nehmen“ und „Geben“ unterschiedlicher Peers zu ermöglichen kann über den Erfolg dieser Technologie entscheiden.

## 3 Forschungsaspekte zu Peer-to-Peer-Systemen und -Anwendungen

Ein wichtiger Forschungsaspekt ist die tiefergehende Untersuchung der Anwendbarkeit des Peer-to-Peer-Paradigmas auf verschiedene *Anwendungstypen*, auch und insbesondere jenseits des Filesharing (Dateitauschbörsen). Im Rahmen des Fachgesprächs „*Qualität in Peer-to-Peer-Netzen*“<sup>2</sup> wurden in bezug zu einem Zeitstrahl verschiedene Peer-to-Peer-Anwendungstypen benannt (Abb. 2). Abbildung 3 beschreibt mögliche Erschwernisse für die Peer-to-Peer-Forschung und deren alltägliche Nutzung. Die Abbildungen 4 und 5 beschreiben einige Forschungsherausforderungen für die nächsten Jahre, wie sie von den Teilnehmern des Fachgesprächs entwickelt wurden. Die zentrale Fragestellung war hierbei, ab welchem Zeitpunkt diese Themen als wichtig erachtet werden.

---

<sup>2</sup>TU Darmstadt, Sept. 2003, <http://www.kom.tu-darmstadt.de/ws-p2p/>

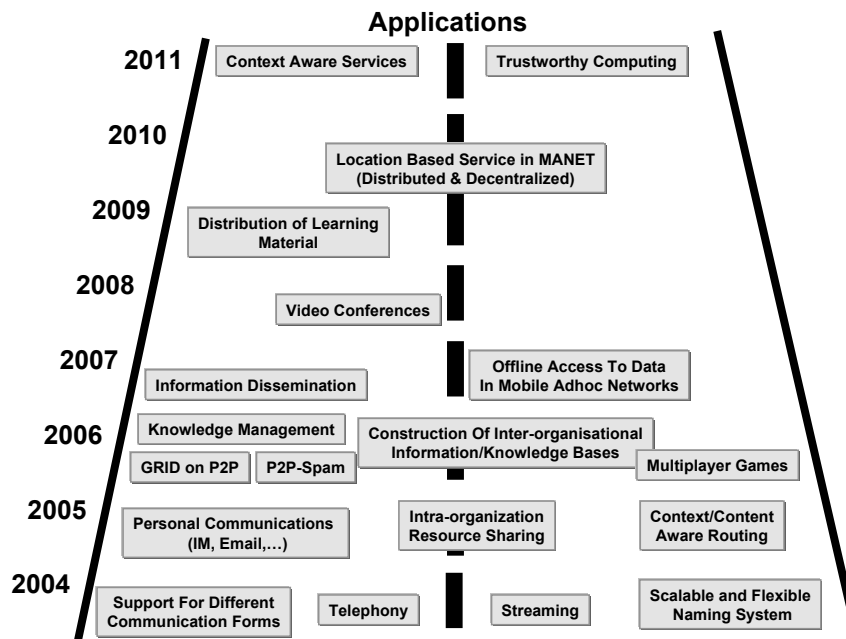


Abbildung 2: Einige Anwendungen (neben dem Austausch von Dateien)

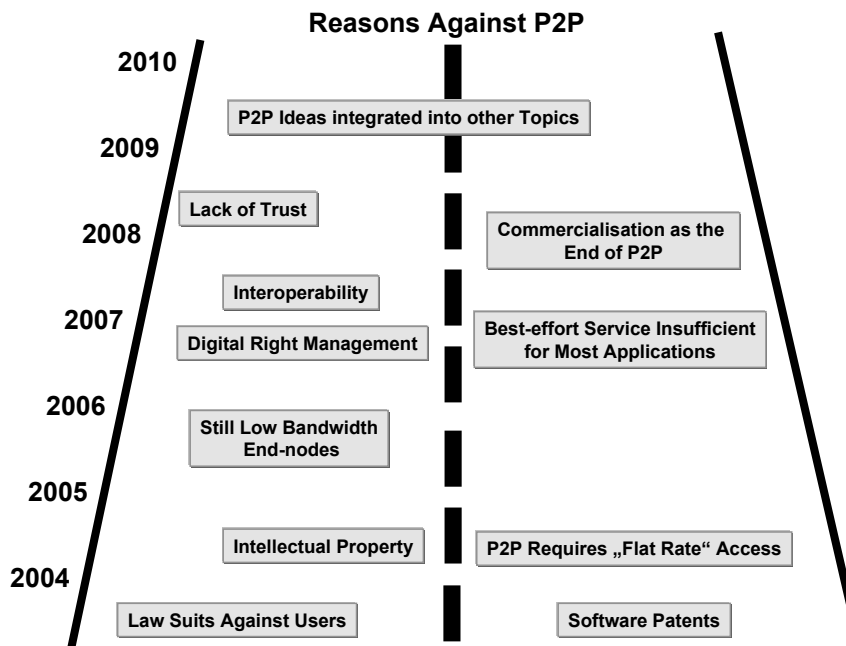


Abbildung 3: Einige Erschwernisse, die einer Verbreitung von P2P-Technologie entgegenstehen.

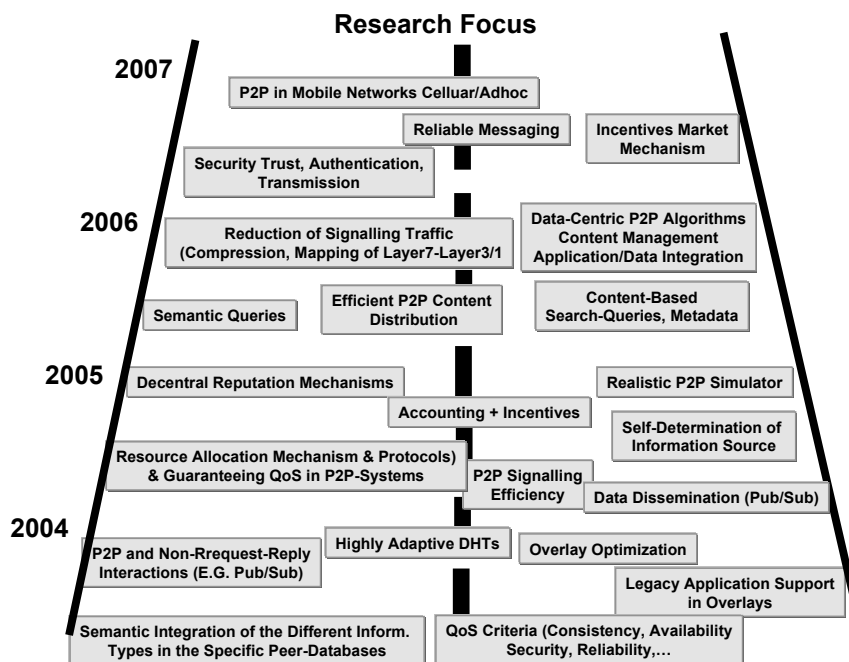


Abbildung 4: Forschungsherausforderungen I (2003 - 2007)

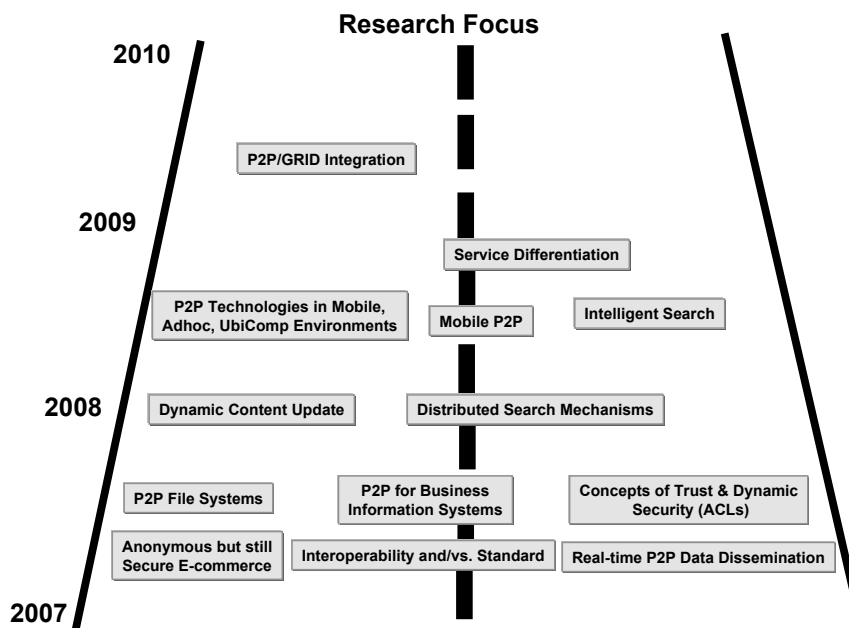


Abbildung 5: Forschungsherausforderungen II (2007 - 2010)

Bei Peer-to-Peer-Systemen besteht eine grundsätzliche Herausforderung in der dezentralen *Selbstorganisation* eines verteilten Systems unter Einhaltung bestimmter Dienstgütern ohne die Nutzung zentraler Dienste. In diesem Zusammenhang kommt der effizienten *Suche* nach der Lokation eines gewünschten Datums und der damit verbundenen Verwaltung der Inhalte eine zentrale Rolle zu. Darauf aufbauend wird ein Großteil der Funktionalität von Peer-to-Peer-Systemen verwirklicht. Im Gegensatz zu zentralen Server-Anwendungen, bei denen der Speicherort eines Datums inhärent bekannt ist, können Daten in dezentralen Architekturen an zahlreichen, zum Teil weit entfernten, Stellen im Netz gespeichert werden. In den vergangenen Jahren haben sich zur Lösung dieses Problems zwei Richtungen entwickelt, die im folgendem kurz vorgestellt werden: *Unstrukturierte* und *strukturierte Peer-to-Peer-Systeme*.

### 3.1 Unstrukturierte Peer-to-Peer-Systeme

Die sogenannten unstrukturierten Verfahren wurden vor allem in den ersten Generationen der Peer-to-Peer-basierten Dateitauschsystemen eingesetzt. Diese Systeme basierten zum Beispiel auf einer Server-orientierten Suche. Dazu wurde die Lokation jedes Datums in einem Server verwaltet und erst nach der Lokation der Daten kommunizierten die Peers direkt untereinander (Hybrider Ansatz vgl. Abb. 1b). Andere Ansätze, wie z.B. Gnutella, nutzen das Prinzip des Flutens, d.h. Suchnachrichten werden an alle beteiligten Peers im System gesendet, bis das gewünschte Datum oder der gewünschte Peer gefunden wird.

Es wird schnell deutlich, dass beide Ansätze nicht skalieren. So besitzen die Server-orientierten Verfahren, neben einem zentralen Angriffspunkt, einen Flaschenhals hinsichtlich ihrer Systemressourcen (Speicher, Rechenleistung und Netzbandbreite) und die Fluten-basierten Verfahren führen zu enormen Belastungen der Kommunikationsverbindungen. In der Regel entstanden diese unstrukturierten Peer-to-Peer-Anwendungen nach Anforderungen des Marktes (größtenteils für Dateitauschbörsen und Instant-Messaging) und wurden dementsprechend schnell und unkoordiniert entworfen.

### 3.2 Strukturierte Peer-to-Peer-Systeme

Von der Forschung wurde die Herausforderung der Skalierbarkeit unstrukturierter Peer-to-Peer-Anwendungen aufgegriffen. So wurden im Hinblick auf die enormen Möglichkeiten dezentraler selbstorganisierender Systeme verschiedene Ansätze zur verteilten, inhalts-adressierbaren Speicherung von Daten (verteilte Indexstrukturen) untersucht. Diese so genannten verteilten Hash-Tabellen (Distributed Hash Tables, DHT) wurden neben dem Aspekt der verteilten Indexierung vor allem mit dem Ziel der Skalierbarkeit, Zuverlässigkeit und Fehlertoleranz entwickelt. So weisen sie gegenüber unstrukturierten Verfahren nicht nur in diesen Punkten, sondern auch in der Leistungsfähigkeit, entscheidende Vorteile auf. Verteilte Hash-Tabellen ermöglichen in der Regel das Auffinden eines Datums im Netz mit einer Komplexität von  $O(\log N)$  - vergleichbar mit den bekannten, nicht-verteilt-

ten Such- und Indexierungsverfahren. Im Vergleich zu den mindestens linearen Komplexitäten der zuvor genannten unstrukturierten Peer-to-Peer-Anwendungen, können sowohl das zugrunde liegende Netz als auch die Anzahl der Teilnehmer beliebig wachsen, ohne die Leistungsfähigkeit der verteilten Anwendung zu gefährden. Notwendige Verwaltungsoperationen, wie etwa das Einfügen neuer Inhalte, neuer Peer-to-Peer-Knoten, aber auch Fehlerfälle, wie der Ausfall von Knoten, besitzen in den meisten Ansätzen verteilter Hash-Tabellen eine Komplexität von  $O(\log N)$  bzw.  $O(\log^2 N)$ .

Die Arbeitsweise der verschiedenen verteilten Hash-Tabellen ist generell sehr ähnlich und unterscheidet sich im Wesentlichen in den verwendeten Such- und Verwaltungsstrategien. So existieren Ring-basierte Ansätze, wie etwa Pastry, Tapestry und Chord, die auf ähnlichen Suchverfahren wie Binärer B\*-Bäume beruhen, und geometrische Verfahren, wie Content Addressable Networks (CAN) oder Viceroy. Generell wird jedem Knoten einer verteilten Hash-Tabelle ein bestimmter Teil des Suchraums  $[0, 2^n - 1]$  zugewiesen, wobei oftmals aus Redundanzgründen eine Replikation in benachbarten Knoten stattfindet. Durch baum-ähnliche Routingverfahren (bspw. bei Pastry) oder Sprunglisten (bspw. bei Chord) wird eine Anfrage von einem beliebigen Einstiegsknoten zum verwaltenden Knoten der gesuchten Kennung weitergeleitet, wobei ein logarithmischer Aufwand gewährleistet werden kann. Die für die Wegewahl notwendigen Informationen umfassen hierbei in der Regel ebenfalls einen Umfang von  $O(\log N)$  Einträgen pro Knoten.

Neben den erwähnten Ähnlichkeiten zu bekannten Indexierungsverfahren aus dem Datenbankbereich ist zu beachten, dass verteilte Hash-Tabellen zusätzlich Mechanismen zur verteilten Verwaltung der Datenstruktur, Verfahren zur Redundanz und dem Auffinden möglichst naher Instanzen des gesuchten Datums verfügen. Hinsichtlich Details der jeweiligen Such- und Verwaltungsoperationen bestimmter verteilter Hash-Tabellen wird auf die jeweiligen Publikationen und auf [BKK<sup>+</sup>03] als gute Übersicht verwiesen.

## 4 Resümee

An künftige verteilte Systeme und Anwendungen werden für einen erfolgreichen Einsatz im Internet gestiegene Anforderungen gestellt. Neben der Skalierbarkeit, einer entsprechenden Qualität und mehr Flexibilität stellen die Verlässlichkeit und der Schutz vor gezielten Angriffen kritische Eigenschaften dar, deren intensive Erforschung mit einer erfolgreichen Umsetzung in die Praxis die Relevanz von Peer-to-Peer-Anwendungen und Systemen entscheidend prägen wird.

## Literatur

- [BKK<sup>+</sup>03] H. Balakrishnan, M.F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Looking up Data in P2P Systems. *Communications of the ACM*, 46(2), February 2003.
- [Ora01] A. Oram. *Harnessing the Power of Disruptive Technologies*. O'Reilly, 2001.
- [SRC84] J.H. Saltzer, D.P. Reed, and D.D. Clark. End-to-end arguments in system design. *ACM Transactions on Computer Systems*, 4(2), 1984.
- [SW04] R. Steinmetz and K. Wehrle. Peer-to-Peer-Networking und -Computing. *GI-Spektrum*, 27(1), February 2004.