Technische Universität Darmstadt

Department of Electrical Engineering and Information Technology
Department of Computer Science (Adjunct Professor)
Multimedia Communications Lab
Prof. Dr.-Ing. Ralf Steinmetz

Qualitätsmerkmale von Peer-to-Peer Systemen

Technical Report

von

Oliver Heckmann, Ralf Steinmetz, Nicolas Liebau, Alejandro Buchmann, Claudia Eckert, Jussi Kangasharju, Max Mühlhäuser, Andreas Schürr

am
3. Juli 2006

KOM-TR-2006-03

Inhaltsverzeichnis

1 Zusammenfassung

Die Bedeutung von Peer-to-Peer (P2P) hat in den letzten Jahren enorm zugenommen. P2P-Anwendungen haben inzwischen vom Verkehrsvolumen im Internet das World-Wide-Web (WWW) als bislang dominante Anwendung verdrängt. Sie basieren auf einem grundsätzlich anderen Kommunikationsparadigma als das WWW und als andere traditionelle (Client-Server) Anwendungen. Hieraus ergibt sich die Frage nach der Leistungsfähigkeit und der Qualität von geeigneten Mechanismen, die auf diesem Paradigma basieren.
In diesem Technischen Report definieren wir die Qualität eines P2P Systems anhand von 4 Qualitätsmerkmalen (Adaptivität, Effizienz, Validität und Vertrauenswürdigkeit) mit zahlreichen Untermerkmalen. Diese Gruppierung erscheint uns aus wissenschaftlicher Sicht sinnvoll für die Untersuchung und Verbesserung der Qualität von P2P-Systemen. Wir verwenden diese Gruppierung selbst in der DFG Forschergruppe QuaP2P (www.quap2p.de).

Verschiedene Communities in der Informatik haben häufig unterschiedliche Definitionen der verwendeten Termini, z.B. Konsistenz oder Kohärenz. Daher haben wir diese Begriffe klar definiert und mit Beispielen unterlegt. Bei der Entscheidung für eine Definition stand im Zweifel eine pragmatische Sicht und das Ziel im Vordergrund, das Qualitätsmerkmal möglichst gut quantitativ messen zu können und möglichst wenig Überlappungen mit anderen Qualitätsmerkmalen zu haben.

Zwischen den unterschiedlichen Qualitätsmerkmalen existiert eine Vielzahl von Abhängigkeiten. Zum Beispiel kann eine Erhöhung der Verlässlichkeit durch das Fluten von Routing-Nachrichten erreicht werden. Dies führt aber gleichzeitig zu einer deutlichen Erhöhung des Kommunikationsaufwands und damit zu einer Verschlechterung der Effizienz des Netzwerks.

In der Forschergruppe QuaP2P (www.quap2p.de) untersuchen wir die in diesem Technischen Report definierten Qualitätsmerkmale und ihre gegenseitigen Abhängigkeiten mit dem Ziel, die Qualität zukünftiger P2P-Systeme zu verbessern. Als Maßstab dienen uns zwei Referenzszenarien, die in [HSL+06] beschrieben sind.

2 Einführung

2.1 Peer-to-Peer Systeme

Ausgehend von der Definition in [SW04b] wird unter einem Peer-to-Peer-System (P2P-System) ein System möglichst gleichberechtigter, autonomer Einheiten (Peers) verstanden, das sich selbst organisiert und 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 Organisation, Ressourcennutzung und Diensterbringung. Die teilnehmenden Einheiten des P2P-Systems bezeichnen wir als Knoten; sie unterhalten wechselseitige Kommunikationsbeziehungen und formen somit ein (P2P-) Netzwerk, das ein Overlay-Netzwerk über das darunter liegende IP-Netzwerk bildet.

2.2 Motivation, Vision

Napster has become the fastest-growing application in the Net's history.

[Shi01, S. 23]

SETI@home has received 200 million results,
*for a total of 4 * 10^{20} floating-point operations.*
We believe that this is the largest computation ever performed.
[And01, S. 49]

Die Bedeutung von P2P-Anwendungen hat in den letzten Jahren enorm zugenommen. Dies zeigt sich zum Beispiel in der schnellen Verbreitung von P2P-Systemen, die mit Anwendungen wie Napster und SETI@home begann. Binnen kürzester Zeit haben sie vom Verkehrsvolumen die zuvor dominierende Anwendung des World-Wide-Web (WWW) verdrängt. So werden beispielsweise in einer Studie von 2003 [AG03b] 50% des Verkehrsvolumens P2P-Dateitauschbörsen zugeschrieben und lediglich 15% dem WWW-Verkehr[1].

Die wachsende Bedeutung von P2P wird auch in der wissenschaftlichen Gemeinde reflektiert: Zum einen durch eine Vielzahl neuer internationaler Konferenzen mit Forschungsschwerpunkt P2P wie z.B. International Conference on Peer-to-Peer Computing (P2P), O'Reilly Peer-to-Peer Conference, International Workshop on Peer-to-Peer Knowledge Management (P2PKM), Workshop on Economics of Peer-to-Peer Systems (P2PEcon), International Workshop on Peer-to-Peer Systems (IPTPS). Zum anderen wurde mit Planet-Lab eine international genutzte P2P-basierte Testumgebung für die wissenschaftliche Untersuchung von P2P-Systemen geschaffen.

P2P ist keinesfalls nur ein Verfahren zum Austausch von Dateien. Mit P2P-Applikationen wie Skype, Edutella und SETI@home etablieren sich P2P-Systeme auch in anderen Anwendungsbereichen. P2P stellt einen Paradigmenwechsel gegenüber dem traditionellen Client-Server-Prinzip dar, von Koordination zu Kooperation, von Zentralisierung zu Dezentralisierung und vom Streben nach Kontrolle hin zu Anreizen [SW04b]. Heute sind die erfolgreichsten P2P-Systeme allerdings die so genannten Dateitauschbörsen [San03, AG03b]. Ihr Erfolg ist maßgeblich auf den überwiegend illegalen Austausch urheberrechtlich geschützter Daten zurückzuführen [HBMS04]. Bezüglich Qualitätsmerkmalen wie der Performanz, Skalierbarkeit, Sicherheit und Zuverlässigkeit zeigen sich in diesem Anwendungsbereich jedoch eklatante Mängel [AAA+03, HK03, WXZ03, MCR03, NW03b, Neu94]. Grundlegend für die Ausweitung des P2P-Paradigmas auf viele weitere Anwendungsfelder sind aber eben solche Qualitätsmerkmale. Ebenso ist das exaktere Verständnis der gegenseitigen Beziehungen und möglicher Zielkonflikte zwischen den Qualitätsmerkmalen unerlässlich.

---

[1] 32% des Verkehrsvolumens konnten nicht eindeutig einer Anwendung zugeordnet werden. Es ist zu erwarten, dass ein großer Teil hiervon ebenfalls P2P-Anwendungen zuzuschreiben ist.

3 Qualitätsmerkmale

„Qualität ist die Gesamtheit von Merkmalen einer Einheit bezüglich ihrer Eignung, festgelegte und vorausgesagte Erfordernisse zu erfüllen.“ [Qua]. Zur Untergliederung der Qualität von Software in Merkmale existieren eine Vielzahl verschiedenster Vorschläge (siehe z.B. [Din, ISO, BBK+78, MWE87, DW88, BKLW95]), die jedoch alle für unsere Ziele wenig zweckmäßig sind. Zum einen, weil dem Kommunikationsaspekt von P2P-Systemen nicht ausreichend Rechnung getragen wird. Beinahe alle für die Qualität von P2P-Systemen relevanten Merkmale werden in den genannten Referenzen unter dem Merkmal „Funktionalität“ subsumiert. Zum anderen stehen Qualitätsmerkmale wie etwa „Benutzbarkeit“, „Wartbarkeit“ oder „Änderbarkeit“ zunächst nicht im Mittelpunkt der Forschung im Bereich Peer-to-Peer-Systeme.

Die vorherrschenden Definitionen der Qualität von Kommunikationsnetzwerken, wie z.B. in ATM[2], RSVP/IntServ[3] und DiffServ[4], sieht eine Bestimmung der Dienstgüte anhand von quantitativen Parametern vor [Sch01a]. Diese können allerdings auch nicht auf die gesamten Qualitätsanforderungen von P2P-Systemen abgebildet werden. Eine Übersicht über weitere Metriken die für Dienstgüte vorgeschlagen wurden, findet sich in [Kal03]. Ein weiterer Ansatz für die Untergliederung der Qualität von Kommunikationsnetzen in Untermerkmale wird in [vM01] präsentiert, der allerdings ebenfalls einige, für P2P-Systeme entscheidende, Aspekte (insbesondere alle Qualitätsaspekte im Rahmen von Overlay-Netzwerken) nicht berücksichtigt. Da augenscheinlich bisher für P2P-Systeme keine Gliederung von Qualität in Qualitätsmerkmale existiert, untergliedern wir die Qualität für P2P-Systeme wie in Abbildung 1: Gliederung der Qualitätsmerkmale beschrieben. Wir unterscheiden vier Qualitätsmerkmale, wobei sich ein Qualitätsmerkmal wiederum in entsprechende Untermerkmale aufgliedern lässt.

Die Ausprägungen jedes Qualitätsmerkmals und Untermerkmals werden durch die in einem P2P-System angewendeten Mechanismen beeinflusst. Da die eingesetzten Mechanismen meist mehrere Qualitätsmerkmale beeinflussen, bestehen starke Abhängigkeiten zwischen den Qualitätsmerkmalen. Um diese zu untersuchen und aufzulösen, betrachtet dieses Projekt zwei, bezüglich der Qualitätsanforderungen weitgehend orthogonale, Referenzszenarien.

Wir unterscheiden die vier Qualitätsmerkmale Adaptivität, Effizienz, Validität und Vertrauenswürdigkeit, wobei sich jedes Qualitätsmerkmal wiederum in entsprechende Untermerkmale aufgliedert:

[2] Asynchronous Transfer Mode [Com99]

[3] Resource Reservation Protocol / Integrated Services [BCS94, BZB+97]

[4] Differentiated Services [BBC+98]

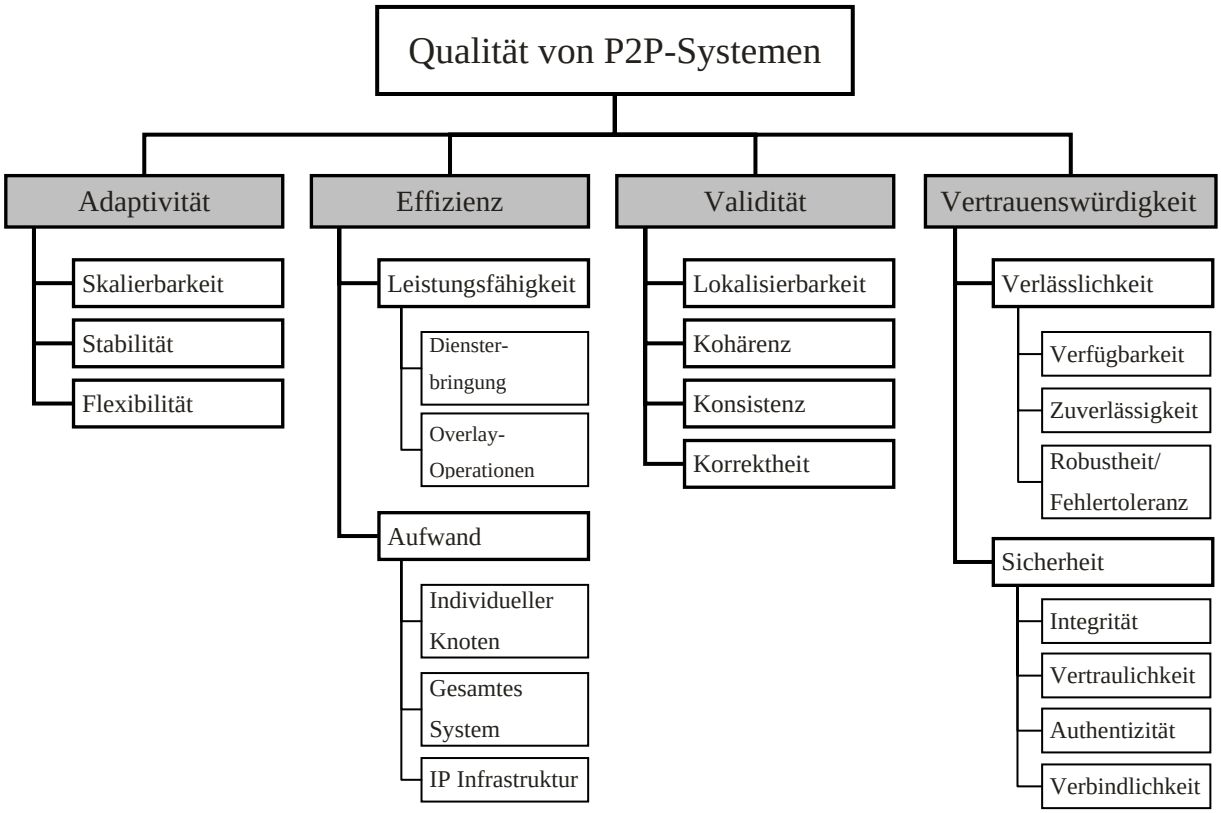

Abbildung 1: Gliederung der Qualitätsmerkmale

3.1 Adaptivität

Bei der Entwicklung von P2P-Anwendungen ist es immer noch gängige Praxis, Software auf dedizierte Anwendungsfälle hin zu spezialisieren und immer wieder weitgehend „von Null an“ zu entwickeln. Nur vergleichsweise elementare Funktionalität wird in Umgebungen wie JXTA bereitgestellt. Wir propagieren den Übergang zu breit einsetzbaren, wieder verwendbaren Komponenten. Offensichtlich ist dieses Ziel nur zu erreichen, wenn die Komponenten und die in ihnen verankerten Methoden und Mechanismen ein hohes Maß an Adaptivität aufweisen: sie müssen in vielerlei Hinsicht anpassungsfähig sein. Wir unterscheiden im vorliegenden Dokument zunächst quantitative und qualitative Adaptivität.

Quantitative Adaptivität bezieht sich offensichtlich auf die Fähigkeit, stark unterschiedliche, ggf. dynamisch variierende Mengen von „Beteiligten“ zu unterstützen, insbesondere Knoten- und Nutzeranzahl. Dieses Untermerkmal bezeichnen wir als Skalierbarkeit.

Als qualitativ fassen wir alle Unterschiede bzw. Veränderungen im „Umfeld“ der Anwendung auf, deren Berücksichtigung sinnvoll ist, z.B. Nutzercharakteristika, Nutzungsorte und -zeitpunkte, verfügbare Basisdienste usw. Dieses Untermerkmal nennen wir Flexibilität.

Erfahrung und Stand der Wissenschaft deuten darauf hin, dass der Vorgang der Anpassung selbst, vor allem wenn er häufig erfolgt, gewünschte Eigenschaften eines Systems gefährden kann. Die Vermeidung solch unerwünschter Nebeneffekte der Anpassung soll mit dem Untermerkmal Stabilität bezeichnet werden. Traditionell wird Stabilität insbesondere im Zusammenhang mit dynamischen quantitativen Anpassungen untersucht, wo die Gefährdung besonders hoch ist; für diesen Teilaspekt existieren dementsprechend die bei weitem elaboriertesten Theorien und verwandten Arbeiten. Auch unsere Forschergruppe will sich mit dedizierten Arbeiten zur Stabilität insbesondere der dynamischen quantitativen Anpassung annehmen.

Es werden daher Skalierbarkeit und Stabilität bezüglich quantitativer Anpassungen verzahnt betrachtet, während Skalierbarkeitsaspekte bezüglich qualitativer Anpassung im Rahmen der Arbeiten zu Flexibilität untersucht werden. Die drei zu untersuchenden Untermerkmale seien nachfolgend nochmals definitionsartig, aber vorläufig und intuitiv erläutert sowie mit je einem Beispiel ergänzt.

- **Skalierbarkeit**: Dieser Begriff bezeichnet die quantitative Anpassungsfähigkeit des Systems, also diejenige an eine sich ändernde Zahl von Entitäten/Knoten (und damit ggf. Nutzern) oder im System zur Verfügung gestellten Diensten.

Beispiel: Ein häuftig zitiertes Beispiel für die mangelnde Skalierbarkeit eines P2PSystems ist das ursprüngliche Gnutella-Netzwerk [Cli00]. Es brach unter den rapide wachsenden Teilnehmerzahlen im Jahr 2000 zusammen [Kil01].

Als Kommunikationsinfrastruktur nutzt die große Mehrheit moderner strukturierter P2P-Systeme Verteilte Hash-Tabellen (DHTs) [BKK+03]. Frühe Ansätze von verteilten Systemen entwickelten verteilte Strukturen basierend auf linearem Hashing [LNS96, LN97]. Im Allgemeinen stellen DHTs eine skalierbare Methode zur Verfügung, um Datenobjekte unter gegebenen Schlüsseln zu adressieren [HKRZ02]. Beim Design von DHTs für P2P-Overlaystrukturen gilt Chord [SMLN+03] als eine wichtige Referenz für etwaige Vergleichsuntersuchungen. Chord konstruiert eine Ringtopologie erweitert um Finger, die als Abkürzung dienen und damit schnelle Nachschlage-Operationen ermöglichen. Um Schlüssel Knoten zuzuweisen benutzt Chord konsistentes Hashing [KLL+97]. Koorde [KK03] ist ein Vorschlag, der das Chord-Design auf de Brujin-Digraphen [dB46] aufsetzt. De Brujin-Graphen bieten eine sehr attraktive Kombination aus asymptotisch optimalem Durchmesser bei gleichzeitig konstantem Knotengrad. Deshalb wurden sie in weiteren Ansätzen verwendet. D2B [FG03] ist ein Netzwerk für adressierbare Inhalte, das de Brujin-Graphen nutzt, um das Overlay zu konstruieren. Omicron [DMS04] ist ein hybrider, DHT-basierter Ansatz basierend auf de Brujin-Graphen, der um Gruppenbildungs- und Rollenmechanismen erweitert ist.

Andere DHT-basierte Konzepte sind beispielsweise Pastry und Tapestry. Pastry [RD01] nutzt einen Prefix-basierten Routing-Mechanismus ähnlich dem Plaxton-Routing[5] zum Aufbau eines selbst organisierten, dezentralen Overlay-Netzwerks. Tapestry [ZHS+04] hat ähnliche Eigenschaften wie Pastry. Es benutzt ein dezentralisiertes zufallsbasiertes Verfahren um sowohl Lastverteilung als auch lokales Routing zu erreichen. Im Gegensatz zu Pastry nutzt Tapestry einen Sufix-basierten Routing-Mechanismus.

Kademlia [MM02] ist ein symmetrisches, DHT-basiertes P2P-Overlay, das eine XOR-basierte Distanzmetrik benutzt, um seine Topologie zu konstruieren und die Ressourcenanzeige Knoten zuzuweisen. Die symmetrische Architektur Kademlias ermöglicht die Nutzung der Suchanfragenachrichten für die Aufrechterhaltung der Overlaystruktur. Dadurch wird der benötigte Aufwand für explizite Signalisierungsnachrichten verringert. Content Addressable Network (CAN) [RFH+01] ist eine weitere verteilte, dezentrale Infrastruktur, die einen multi-dimensionalen kartesischen Koordinatenraum auf einen multidimensionalen Körper abbildet. Skipnet [HJS+03] und SkipGraph [AS03] sind zwei sehr ähnliche strukturierte Overlay-Netzwerke (obwohl sie unabhängig von einander entwickelt wurden), die Auslassungslisten (skip lists) [Pug90], eine probabilistische Datenstruktur, erweitern. Sie sind ähnlich zu Chord mit dem wesentlichen Unterschied, dass die Finger nicht mehr exponentiell verteilt sein müssen (Finger sind hier zufällig platzierte Abkürzungen). Viceroy [MNR02] ist ein strukturiertes Netzwerk basierend auf einer Schmetterlingstopologie (Butterfly-Topologie). Die Knoten in einem Viceroy-Overlay benötigen mit hoher Wahrscheinlichkeit lediglich eine konstante Anzahl von Nachbarn während der Durchmesser des Graphen logarithmisch

[5] Plaxton et al. [PRR97] schlagen eine verteilte Datenstruktur vor, bekannt unter dem Namen „Plaxton Mesh“. Es ist für ein Netzwerk-Overlay optimiert, das das Auffinden von benannten Datenobjekten unterstützt, die zu einem Wurzelknoten verknüpft sind.

zunimmt. AGILE (Adaptive, Group-of-Interest-based Lookup Engine) [MS03] ist ein DHT-basiertes, strukturiertes Overlay-Netzwerk, das die Gemeinsamkeiten in den Interessen der Benutzer berücksichtigt, um ein effizientes System zu entwerfen.

Neben der eben erläuterten Auswahl strukturierter Overlay-Netzwerke gibt es einige unstrukturierte Ansätze. Die Skalierbarkeitseigenschaften unstrukturierter Overlays wurde bereits in einigen Arbeiten untersucht. Lv et al. [LCC+02] diskutieren skalierbare Suchmethoden wie z.B. erweiterbare Ringe und mehrfache Zufallsbewegungen. Ebenso untersuchten Lv et al. wie Peer-Heterogenität genutzt werden kann, um Systeme wie Gnutella besser skalierbar zu gestalten als im homogenen Fall [LRS02]. Yang und Garcia-Molina [YGM02] erforschten die Leistungsfähigkeit von gerichteten Breitesuchetechniken und lokalen Indizes. Oceanstore [KBC+00] nutzt hybride Techniken basierend auf Tapestry und abgeschwächten Bloom-Filtern [Blo70] um populäre Objekte mit hoher Wahrscheinlichkeit effiiient zu finden [RK02]. Adamic et al. [ALPH01] untersuchen skalierbare Suche in Power-Law-Netzwerken. Small-World-Netzwerkeigenschaften [Mil67] wurden ebenso im Kontext von P2P untersucht [Ada99]. Zhang et al. [ZGG02] nutzen die Small-World-Eigenschaften um die Leistungsfähigkeit von Freenet [CSWH00] zu steigern. SWAN [Bon02] ist ein P2P-Netzwerk, das auf dem Small-World-Phenomen aufbaut. Ähnlich basiert auch Symphony [MBR03] auf dem Small-World-Phenomen. Hierarchische Ansätze wie eDonkey [Tut04, HLD+05] oder das FastTrack-Protokoll [Fas], das in KaZaA [LKR04] genutzt wird, nutzen das Konzept von Super-Peers (und Servern) in P2P-Architekturen, um die Suchanfrage-Routing-Kosten zu reduzieren. Allerdings können Server in solchen Ansätzen potentielle Flaschenhälse werden, wenn sie nicht mit der gesamten Netzwerkgröße gut ausbalanciert werden.

- **Stabilität**: Als Stabilität sei die Fähigkeit eines P2P-Systems bezeichnet, bei geänderten Rahmenbedingungen, insbesondere bei häufigen Abfolgen von Adaptionsvorgängen, seine Funktionalität aufrechtzuerhalten.

Beispiel: Komplexe verteilte Systeme können durch sich ändernde Rahmenbedingungen in Schwingung geraten; ein berühmtes Beispiel sind die Routing-Instabilitäten der Interior- und Exterior-Routing Protokolle des Internets, siehe z.B. [BR01]. Auch beim Routing in P2P-Overlaynetzen ist ähnliches Verhalten zu erwarten und muss deshalb untersucht werden. Stabilität bezieht sich vordringlich auf quantitative Adaption. Stabilität qualitativer Adaption wird nach derzeitigem Kenntnisstand als Bestandteil von Kontextbewusstsein untersucht.

Im Kontext von P2P-Systemen haben Stabilitätsuntersuchungen in der Forschung bisher noch nicht die gleiche Aufmerksamkeit erhalten wie Skalierbarkeits-, Lastverteilungs- und Fehlertoleranzuntersuchungen. Es gibt allerdings einige interessante erste Ansätze. Z.B. diskutieren Omicron [DMS04] und Coral [FFM04] Stabilitätsmechanismen, um das Oszillieren beim Aufbau und Zerteilen von Gruppen von Knoten und beim Migrieren von Knoten von Gruppe zu Gruppe zu vermeiden. Weiterhin definiert Chord [SMLN+03] eine Stabilisierungsperiode, um die Finger zu korrigieren, die beim Beitreten und Verlassen von Knoten ungültig werden. Die stochastische Analyse von Chord findet sich in [BSH05]. Netzwerkstabilität wird ebenfalls in [DMS05] diskutiert, wo eine bedingte Zuverlässigkeitsmethode genutzt wird, um die zuverlässigsten Knoten zu identifizieren und das Netzwerk um sie herum aufzubauen, um damit eine stabile Overlay-Topologie zu erhalten. YAPPERS [GSGM03] ist ein hybrider P2P-Ansatz, der Datenstabilität trotz dynamischer Teilnahme der Knoten am Netzwerk erreicht. Kelips [GBL+03] erreicht Stabilität (Ausfallsicherheit) beim Beitreten und Verlassen von Knoten, indem es ein epidemisches Multicast-Protokoll nutzt, um Mitgliedschaften und indizierte Daten zu replizieren.

- **Flexibilität**: Unter Flexibilität sei die Anpassbarkeit eines P2P-Systems an veränderliche äußere ("qualitative") Umstände verstanden. In der Forschung zu "context aware computing" wird die Fähigkeit von Softwaresystemen erforscht, den Kontext der Nutzung zu berücksichtigen, d.h. diesen als zusätzliche Eingabe aufzunehmen und intern sowie in der Wirkungsweise inkl. Benutzeroberfläche zu reflektieren. Diese Forschung soll erstens auf P2P-Systeme angepasst, zweitens im Sinne von

Flexibilität verallgemeinert und drittens mit den anderen Qualitätsmerkmalen harmonisiert werden. In Erweiterung des in der "context awareness"-Forschung verwendeten Kontextbegriffes sollen insbesondere Nutzerpräferenzen und verfügbare Basissysteme betrachtet werden. Ein Beispiel: P2P-Konzepte bieten sich für Softwareanwendungen für so genannte „community networks“ (Netzwerke von Personen mit ähnlichen Interessen) besonders an, deren Nutzer über spontane Mobilkommunikation verbunden sind: diese Nutzer verhalten sich nämlich dann "automatisch" sehr ähnlich den Peers im P2P-Sinne. Flexibilität in solchen Systemen betrifft unter anderem die Anpassbarkeit an den gegenwärtigen oder potenziellen Aufenthaltsort - im erwähnten Forschungsbereich als "Lokationskontext" bezeichnet. Dieser Lokationskontext beeinflusst hochdynamisch die physische Verbindungstopologie, die verfügbare Bandbreite und die Anwendung selbst (z.B. sinnvolle Informationsangebote). Weitergehende Anpassungsfähigkeit könnte Medientypen, von Nutzern geforderte Kohärenz der verteilt angebotenen Informationen, geforderte und erhältliche Sicherheitsgarantien und vieles mehr betreffen.

3.2 Effizienz

In Anlehnung an die Definition in [Wiki] definieren wir Effizienz als das Verhältnis einer in definierter Qualität vorgegebener Leistungsfähigkeit zu dem Aufwand, der zur Erreichung der Leistungsfähigkeit nötig ist:

$$Effizienz = \frac{Leistungsfähigkeit}{Aufwand}$$

Die Leistungsfähigkeit untergliedern wir in zwei Untermerkmale:

- Diensterbringung, die in P2P-Systemen direkt zwischen den (üblicherweise zwei) beteiligten Knoten stattfindet, also „Peer-to-Peer“.

Beispiel: Bei Dateitauschbörsen ist die eigentliche Diensterbringung der Transfer der getauschten Datei oder eines Dateifragments. Auch bei anderen Sorten von Diensten in P2P-Systemen kann die Diensterbringung aus Netzwerksicht letztlich auf einen Datentransfer zurückgeführt werden.

- Overlay-Operationen, die zum einen
  - aus den Operationen zur Verwaltung und Aufrechterhaltung des P2P-Overlays und zum anderen
  - aus den Routingoperationen von Nachrichten des P2P-Systems im Overlay-Netzwerk bestehen.

Beispiel: Bei (strukturierten wie unstrukturierten) P2P-Overlay-Netzwerken werden Such- und Look-Up-Nachrichten durch das Overlay-Netzwerk weitergeleitet. Es gibt eine Vielzahl unterschiedlicher Methoden dies durchzuführen, beispielsweise finden Flooding, Random-Walk, Bloom-Filter und Fingertabellen Anwendung.

Da verschiedene Interessengruppen betroffen sind, muss der verursachte **Aufwand** – wie die Praxis gezeigt hat - aus verschiedenen Blickwinkeln beurteilt werden [AG03b, LRS02, Kil01, HBMS04, Hei03]. Aus unserer Erfahrung heraus schlagen wir folgende Blickwinkel vor:

- Sicht des *gesamten P2P-Systems*: Die dem gesamten System verfügbaren Ressourcen müssen offensichtlich effizient genutzt werden, z.B. weil das System sonst einem vom Funktionalität vergleichbaren aber effizienteren System unterlegen wäre.

- Sicht eines individuellen Knotens bzw. Benutzers: Auch wenn das P2P-System als Gesamtsystem hocheffient arbeitet, ist es möglich, dass der Aufwand für einen einzelnenKnoten/Benutzer

unakzeptabel hoch ist oder er sogar überlastet wird. Daher ist auch die Sicht der einzelnen Knoten unbedingt zu berücksichtigen.

- Diese beiden Blickwinkel reichen allerdings nicht aus. Es sind auch die Auswirkungen auf die unterstützende IP-Infrastruktur zu betrachten. Ein P2P-System darf nicht die Gesamteffizienz der darunter liegenden IP-Infrastruktur gefährden; es darf sich also nicht zu sehr auf die Performanz anderer Anwendungen (bzw. auf bestimmte andere Anwendungen) im gleichen IP-Netzwerk - Internet - auswirken[6]. Das ist insbesondere wichtig, da P2P-Systeme bereits heute den volumenmäßig größten Anteil am gesamten Verkehrsaufkommen des Internet haben [AG03b].

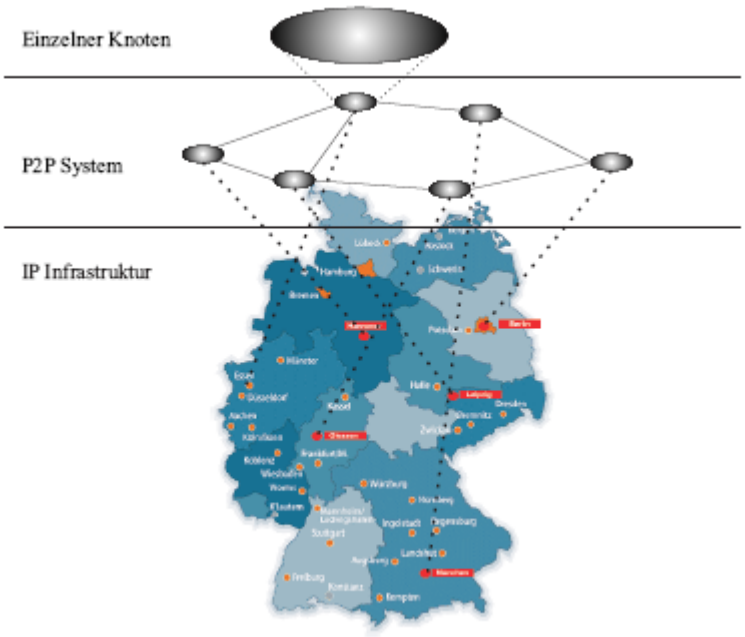

Abbildung 2: Blickwinkel zur Beurteilung des Aufwands

Beispiel: Die Effizienz eines P2P-Systems lässt sich zum Beispiel für den individuellen Teilnehmer verbessern, wenn man wie im Beispiel der Dateitauschbörse eDonkey [HBMS04] UDP-Nachrichten zur Unterstützung der Suche verwendet. Dies kann allerdings bei massivem Einsatz negative Auswirkungen auf den restlichen Verkehr in der IP-Infrastruktur mit sich ziehen, da diese UDP-Nachrichten keiner Fluss- und Überlastkontrolle unterliegen und nicht TCP-freundlich (TCP friendly) sind.

[6] Eine in groben Zügen vergleichbare Problematik wird seit der Entstehung von Multimediakommunikationsdiensten wie Sprach- und Videodiensten unter dem Stichwort TCP-Freundlichkeit (TCP friendliness) diskutiert: Da diese Dienste ursprünglich Kommunikationsprotokolle verwendeten, die sich nicht wie TCP an die Überlastsituation angepasst haben, konnten sie die Leistungsfähigkeit von TCP-basierter Kommunikation stören. Bei P2P-Protokollen liegt die Problematik allerdings weniger in dem Verhalten bei Überlast als in dem Verkehrsaufkommen.

3.3 Validität

Das Qualitätsmerkmal Validität der in einem P2P-System gespeicherten Daten umfasst verschiedene Untermerkmale, die insbesondere unter der Möglichkeit von Updates zum Tragen kommen. Wenn die Daten oder gespeicherten Einheiten in einem P2P-System keinen Änderungen unterworfen und für die interne Struktur der betrachteten Daten keine Einschränkungen (Constraints, Integritätsbedingungen) festgelegt sind, dann sind praktisch alle Systeme in der Lage, bei genügend hoher Anzahl von Replikaten die Auffindbarkeit von „gültigen“ Daten im P2P-System zu gewährleisten. Anders ist es jedoch, wenn man Updates der Daten im P2P-System erlaubt und unterstützen will. Dabei muss man die Art der Updates unterscheiden: additive Updates, die eine neue Version erzeugen, und destruktive Updates, die die alten Werte ersetzen. Additive Updates, die neue Versionen erzeugen sind aus dem CAD- und Document-Management-Umfeld bekannt. Destruktive Updates entsprechen der klassischen Datenbanksemantik von Updates, wo der neue Wert den alten ersetzt. Dementsprechend fassen wir unter dem Qualitätsmerkmal Validität die Untermerkmale Lokalisierbarkeit, Kohärenz, Konsistenz und Korrektheit:

- Die **Lokalisierbarkeit** von Daten ist stark von der Granularität sowie der bekannten Semantik (der internen Struktur) der Daten abhängig. Ursprünglich wurden in P2P-Systemen große Objekte ohne bekannte Semantik (opaque objects) gespeichert und über eindeutige Identifikatoren gesucht, z.B. Musikdateien über ihren Datei-Hash. Je feingranularer die Objekte im P2P-System sind, umso wichtiger wird die Semantik dieser Objekte, z.B. um Queries mit Prädikaten auswerten zu können oder Dokumente über deren Inhalt zu lokalisieren. Die Lokalisierbarkeit von Daten hängt daher von der Semantik der Daten, d.h. den Metadaten und der Struktur der gespeicherten Objekte (z.B. einfache Schlüssel-Wert-Paare, Tupel einer horizontal fragmentierten Relation oder strukturierte Objekte), von der Art der Indexierung sowie der Qualität der Indexwartung bei sich ändernden Datenbeständen und instabilen Knoten ab. Die Lokalisierbarkeit von Daten muss auch in Relation zum Suchaufwand betrachtet werden, da einige Protokolle die Tiefe einer Suche über die Lebenszeit einer Anfrage (time to live, TTL) oder die Anzahl der besuchten Knoten (hop-count) einschränken. Die Lokalisierbarkeit von Daten wird in einem P2P-System über die Anzahl und die Platzierung der Replikate kontrolliert.

- Daten im P2P-System weisen Aktualität (freshness) auf, wenn die Ergebnismenge einer Suche die letzte (aktuellste) Version der Daten enthält. Oft wird in diesem Zusammenhang auch von Kohärenz der zur Verfügung gestellten Daten gesprochen, wenn für jeden Zugriff deren Aktualität garantiert wird. In einem P2P-System ohne zentrale Kontrolle kann die Kohärenz der Daten in verschiedener Weise beeinträchtigt werden. Zum Beispiel kann eine neue Version im P2P-System existieren; diese ist jedoch noch nicht propagiert worden bzw. der dazugehörige dezentrale Index wurde noch nicht aktualisiert. Knoten, die eine ältere Version speichern, antworten mit einer alten Version ohne zu wissen, dass diese nicht mehr aktuell ist und Knoten mit alten (Teil-) Indizes verweisen auf alte Versionen. Die Kohärenz kann über Ablaufzeiten bzw. Gültigkeitsintervalle kontrolliert werden. Für bestimmte Anwendungen (siehe Szenario B) kann auch kontrollierbare Kohärenz, z.B. der Zugriff auf eine bestimmte Version oder Versionen mit bestimmten Eigenschaften, wünschenswert sein.

- Hinter dem Untermerkmal Konsistenz verbirgt sich die Forderung, dass alle Replikate von gespeicherten Objekten bzw. materialisierten Sichten übereinstimmen. Es handelt sich damit um eine Verschärfung der oben aufgestellten Kohärenzforderung. Bei destruktiven Updates wird erwartet, dass ein einheitlicher, konsistenter Wert in allen Replikaten enthalten ist, im Sinne der Datenbankkonsistenz (single copy consistency). Bei additiven Updates ist sicherzustellen, dass die Metadaten der betrachteten Objekte, die deren Versionsgeschichte beschreiben, im obigen Sinne konsistent gehalten werden. Datenbankkonsistenz wird über Quorum-Protokolle in verteilten Systemen erreicht, die mit Einschränkungen auch für P2P-Systeme angewandt werden können. Allerdings wirkt sich die extrem hohe Anzahl von Replikaten, die wegen der hohen Ausfallrate der Knoten nötig ist, auf Quorum-Protokolle negativ aus. Diese sind daher prinzipiell auf P2P-Systeme mit hochverfügbaren Knoten limitiert, ein typischer Fall der Verknüpfung von zwei oder mehr Qualitätsmerkmalen. Für P2P-Systeme mit hoher Ausfallrate werden schwächere Konsistenzgrade

gefordert, wie z.B. gelegentliche Konsistenz (eventual consistency), bei der nur garantiert wird, dass Updates irgendwann auf allen Kopien ausgeführt werden, oder es werden probabilistische Konsistenzkriterien unterstützt. Bei allen Konsistenzkriterien ist der Zeitpunkt des Eintritts bzw. die Wiederaktivierung eines Knotens im P2P-System kritisch, da dieser sich über vergangene Updates informieren muss. Optimistische Verfahren mit unterschiedlichen Abgleichstrategien werden sowohl für verteilte Dateisysteme als auch für mobile Anwendungen verwendet.

- Das Untermerkmal Korrektheit bezieht sich auf die Wahrung von Regeln (integrity constraints), die sich auf die Struktur der betrachteten Daten beziehen und damit deren (statische) Semantik festlegen. Es handelt sich hierbei also um den relativen Korrektheitsbegriff der Informatik, der die Übereinstimmung eines Dokumentes (Programmes) mit seiner Spezifikation, einer Menge von Regeln, fordert. Für diese Form der Korrektheit von Daten werden oft auch die Begriffe „Integrität“ oder „Konsistenz“ (z.B. im IEEE-Standard 830) verwendet, die aber in dem hier betrachteten Umfeld bereits mit anderen Qualitätsmerkmalen assoziiert werden. Die Betrachtung der Korrektheit von Daten ist auf P2P-Systeme beschränkt, die schemabasiert sind oder die Möglichkeit bieten, die Beziehungen zwischen Dokumenten oder Teilen von Dokumenten zu beschreiben. Hierbei ist neben der internen Korrektheit (interne Integrität) einzelner strukturierter Datenobjekte vor allem die externe Korrektheit (externe Integrität) von Interesse. Externe Korrektheit bezeichnet in diesem Zusammenhang die Überwachung von Bedingungen und Beziehungen zwischen verschiedenen strukturierten Datenobjekten, die getrennt voneinander geändert oder versioniert werden können. Die interne Korrektheit bezieht sich in Folge dessen auf die Überwachung von Regeln, die sich auf die interne Struktur eines einzelnen ggf. versionierten Datenobjektes bezieht. Beide Formen der Korrektheit treten beispielsweise bei Dokumentmanagementsystemen oder Softwarekonfigurationsmanagementsystemen auf, die strukturierte Dokumente mit Querverweisen verwalten und einer Versionierung unterwerfen.

Beispiel**:** In einem geographisch über mehrere Organisationseinheiten verteilten Softwareentwicklungsprojekt besteht eine der großen Herausforderungen darin, die Validität der erzeugten versionierten Entwicklungsdokumente wie Lastenhefte, Designdokumente, Quellcode oder Testpläne an allen beteiligten Standorten zu garantieren. Lokalisierbarkeit bedeutet in diesem Zusammenhang, dass jeder Entwickler zu jedem Zeitpunkt Zugriff auf alle erforderlichen Dokumente aller anderen Entwickler hat, während Kohärenz bzw. Konsistenz zusätzlich garantiert, dass kein Entwickler mit veralteten Versionen von Dokumenten (unwillentlich) arbeitet. Korrektheit schließlich befasst sich mit der Fragestellung, ob beispielsweise die gerade betrachtete Version eines Testplans alle in der zugehörigen Lastenheftversion geforderten Abnahmetests umsetzt, für jede in der entsprechenden Designdokumentversion eingeführte Komponente eine Anzahl von Black-Box-Tests enthält oder bezüglich der aktuellen Quellcodeversion und einem gewählten Überdeckungskriterium eine hinreichende Anzahl von White-Box-Tests aufführt.

3.4 Vertrauenswürdigkeit

Unter dem Qualitätsmerkmal der Vertrauenswürdigkeit verstehen wir die Verlässlichkeit und Sicherheit eines P2P-Systems. Maßnahmen zur Gewährleistung der Sicherheit befassen sich mit der Abwehr von Angriffen, die in der Regel gezielt herbeigeführt werden, um beispielsweise Daten zu manipulieren oder nicht autorisiert Kenntnis von sensiblen Informationen zu erlangen. Demgegenüber befasst sich das Gebiet der Verlässlichkeit mit Maßnahmen zur Lösung von Problemen, die zufällig auftreten und nicht bewusst hervorgerufen werden. Verlässlichkeit und Sicherheit sind keine absoluten, atomaren Eigenschaften. Sie besitzen jeweils unterschiedliche Untermerkmale, wie Verfügbarkeit, Zuverlässigkeit, Robustheit, Informationsvertraulichkeit oder Datenintegrität. Auf die relevanten Untermerkmale gehen wir im Folgenden kurz ein.

In einem P2P-System, das das Qualitätsmerkmal der Sicherheit erfüllt, muss gewährleistet sein, dass die Dienste und Objekte, die das System den Nachfragern zur Verfügung stellt, authentisch, integer und ggf. vertraulich und verbindlich sind:

• Unter dem Untermerkmal *der Datenintegrität* versteht man die Eigenschaft eines Systems, ein unbemerktes und unautorisiertes Manipulieren von Daten zu verhindern. Beispielsweise könnte ein Angreifer den Netzwerkverkehr abhören und die darin enthaltenen Daten ändern, weitere hinzufügen oder Daten entfernen. Ein Angreifer könnte auch versuchen, die lokal auf einem Knoten gespeicherten Daten zu modifizieren. Dies könnte durch Einschleusen von fremdem Code durch einen Trojaner, durch einen Buffer-Overflow-Angriff geschehen oder auch direkt durch nicht vertrauenswürdige Nutzer, die die Daten auf ihren Peer-Rechnern absichtlich manipulieren. Durch Maßnahmen zur Sicherung der Integrität (u.a. Hash-Funktionen und Zugriffskontrollen) müssen derartige Angriffe abgewehrt werden.

• Neben der Integrität wird es häufig auch erforderlich sein, die *Vertraulichkeit* der Inhalte zum einen während des Transports über unsichere Netze und zum anderen auch bei ihrer Verarbeitung auf den jeweiligen Peers dezentral zu garantieren. Das Untermerkmal der Vertraulichkeit besagt, dass unautorisierte Dritte keine Kenntnis von sensiblen Informationen erlangen dürfen. Eine verschlüsselte Datenübertragung reicht zur Gewährleistung dieser Anforderung sicherlich nicht aus, da auch die Informationsweitergabe zwischen Peers sowie der Zugriff auf die gespeicherte Information auf den Peers so zu regulieren ist, dass die Vertraulichkeit gewahrt bleibt. Für P2P-Systeme ergeben sich somit Fragen des Information-Rights-Managements, die über die klassischen Lösungen für die Informationsvertraulichkeit wie IPSEC oder SSL als sichere Kommunikationsprotokolle weit hinausgehen.

• Um ein Accounting und Billing für in Anspruch genommene Dienste bzw. genutzte Objekte durchführen zu können, müssen sich die Partner (Peers bzw. einzelne Nutzer) wirksam authentifizieren und die in Anspruch genommenen Dienstleistungen müssen verbindlich einem Nutzer zuzuordnen sein. Unter dem Untermerkmal *Authentizität* eines Objekts bzw. Subjektes verstehen wir die Echtheit und Glaubwürdigkeit des Objekts bzw. Subjekts, die anhand einer eindeutigen Identität und charakteristischer Eigenschaften überprüfbar ist. In einem P2P-Szenario müssen Konzepte bereit stehen, so dass sich sowohl die menschlichen Nutzer als auch die Knoten, die Dienste anbieten sowie die Dienste selber authentifizieren können. Damit eine Kommunikation auch spontan erfolgen kann, werden unterschiedliche Vertrauensmodelle für P2P-Systeme benötigt, die eine entsprechende Flexibilität unterstützen. Um die Authentizität von Kommunikationspartnern sicherzustellen, verwendet man üblicherweise Challenge-Response-Verfahren auf der Basis symmetrischer oder asymmetrischer Verschlüsselung. Hash-Funktionen und digitale Signaturen werden eingesetzt, um den authentischen Ursprung von Daten nachzuweisen. Für große P2P-Systeme ist hierbei u.a. die Frage des Schlüsselmanagements und des Managements der Zugriffsberechtigung (Credentials) zu lösen.

• Unter dem Untermerkmal der *Verbindlichkeit* versteht man, dass es für bestimmte Aktionen nicht möglich ist, diese nach deren Durchführung abzustreiten. In einem P2P-System könnte ein Knoten versuchen, Leistungen eines anderen Knoten in Anspruch zu nehmen und anschließend diese Aktionen abstreiten. Auch könnten legitime Nutzer eines P2P-Systems bewusst versuchen, in Anspruch genommene Dienste abzustreiten. Besonders relevant ist dies, wenn für gewisse Aktionen eine Gegenleistung oder eine Bezahlung erwartet wird. Herkömmlicherweise werden digitale Signaturen zur Lösung der Verbindlichkeitsanforderungen eingesetzt. Diese erfordern aber eine Public Key Infrastruktur (PKI) als zugrunde liegende Infrastruktur, was für P2P-Systeme problematisch ist. Für P2P-Systeme sind alternative Mechanismen notwendig, die flexibel einsetzbar und dennoch den gewünschten Grad an Verbindlichkeit garantieren. Über diese klassische Definition des Untermerkmals Verbindlichkeit hinausgehend, sind für P2P-Szenarien Erweiterungen erforderlich. Von Bedeutung ist hierbei zum Beispiel die Gewährleistung der Rechtzeitigkeit von Aktivitäten oder die Zeitgebundenheit von Aktionen, z.B. wenn ein Angebot einer Dienstleistung nur für einen gewissen Zeitraum verlässlich aufrechterhalten werden kann.

• Neben der eindeutigen Identifizierung von Nutzern ist jedoch gleichzeitig dafür zu sorgen, dass die *Privatsphäre* und die *Anonymität* des Nutzers gewahrt bleiben. Dies ist ein Beispiel von einander ggf. entgegenstehenden, mehrseitigen Sicherheitsinteressen (z.B. Nutzer und Anbieter), die in einem P2P-System aufeinander abzustimmen sind. Zur Wahrung der Privatsphäre ist dafür zu sorgen, dass keine unautorisierten Zugriffs- bzw. Nutzungsprofile erstellt werden können und dass

Profilinformationen über Nutzer, die seine Privatsphäre berühren (z.B. seine Präferenzen, seine Kontaktdaten), nicht nur während der Kommunikation, sondern auch auf den Peers, die diese Information erhalten, vertrauenswürdig und vertraulich verwaltet werden.

Zur Gewährleistung der angesprochenen Untermerkmale der Sicherheit sind die traditionellen Ansätze nicht ohne weiteres auf P2P-Systeme übertragbar, da zentrale Kontrollen und vorab installierte Infrastrukturen in einem dezentralen, sich weitestgehend flexibel selbst organisierenden System nicht anwendbar sind. Es müssen neue Sicherheitsmechanismen entwickelt werden, um sicher zu stellen, dass die Schutzanforderungen (Integrität, Vertraulichkeit, Authentizität) für schützenswerte Güter, die über das P2P-System verteilt werden, auch dezentral gewährleistet werden.

Als wesentliche Untermerkmale werden unter dem Qualitätsmerkmal Verlässlichkeit die Untermerkmale Verfügbarkeit, Zuverlässigkeit und Robustheit subsumiert:

- Die *Verfügbarkeit* (Availability) bezeichnet die Fähigkeit eines P2P-Systems, Dienste/Informationen auf Anforderung zu liefern. Die Knoten der P2P-Systeme werden zumeist aus den Endsystemen der Benutzer geformt. Da Endsysteme abgeschaltet sein können und sich unter wechselnden IP-Adressen dem Netz gegenüber ausweisen können, sind einzelne Knoten einer sich verändernden Konnektivität unterworfen. Die Dienste des P2P-Systems werden von den einzelnen Knoten bereitgestellt. Die Verfügbarkeit bestimmter Dienste im P2P-System hängt damit maßgeblich von der Anzahl der Knoten, die diesen Dienst bereitstellen können, und deren Konnektivitätseigenschaften ab. Durch geeignete Replikationsmechanismen lässt sich daher die Verfügbarkeit als ein Unterpunkt der Verlässlichkeit beeinflussen [OSS03, On04].

Ein Beispiel für Verfügbarkeit in einer bereits breit eingesetzten P2P-Anwendung ist die Art, wie Skype Benutzerinformationen verwaltet. Skype verwendet dazu eine Supernode-Architektur; dabei speichert jeder Supernode Informationen über die Peers und Benutzer, die zu ihm verbunden sind. Die gespeicherten Metadaten (wie etwa Benutzername und IP-Adresse) enthalten dabei die Information, die notwendig ist, um die Verfügbarkeit eines Benutzers zu bestimmen. Ist ein Benutzer verfügbar, so ist es prinzipiell möglich, ihn anzurufen. Dabei muss allerdings berücksichtigt werden, dass andere Faktoren diese Möglichkeit beeinträchtigen können, etwa, wenn die verfügbare Netzwerk-Bandbreite zu gering ist. Die Möglichkeit, den Benutzer tatsächlich zu erreichen, ist Teil der Zuverlässigkeit.

- *Zuverlässigkeit* (Reliability) wird wie folgt definiert: Zuverlässigkeit ist die Fähigkeit des Systems, die Informationen und Dienste so wie spezifiziert zu liefern. Die Hauptschwierigkeit liegt dabei in der Spezifikation eines Dienstes. So könnte zum Beispiel ein Dienst, der Dateien speichert, wie folgt spezifiziert sein: Akzeptiere eine Datei zum Speichern und stelle sie später wieder bereit. Ein Dienst der diesen beiden Kriterien nicht entspricht, würde nicht als zuverlässig gelten. Zuverlässigkeit ist eine stärkere Eigenschaft als Verfügbarkeit, da Verfügbarkeit eine notwendige Vorbedingung für Zuverlässigkeit ist. Kann ein System einen Dienst nicht verfügbar machen, so spielt die Zuverlässigkeit dieses Dienstes keine Rolle mehr, da sie auch nicht überprüft werden kann. Der Grund, warum wir diese beiden Aspekte getrennt behandeln, ist, dass es manchmal ausreicht zu wissen, ob ein spezieller Dienst oder eine Dienstklasse verfügbar ist oder nicht.

In vielen Fällen kann Zuverlässigkeit genauso wie Verfügbarkeit durch Replikation erzielt werden. Die dabei replizierten Daten sind allerdings verschieden: Bei der Verfügbarkeit wird die Information dupliziert, die erlaubt, die Existenz eines Dienstes festzustellen; bei der Zuverlässigkeit wird dagegen der Dienst selbst repliziert. So kann zum Beispiel ein P2P-Telefoniedienst seine Verfügbarkeit dadurch steigern, dass er die Information, ob ein Benutzer mit dem System verbunden ist, repliziert. Zuverlässigkeit ist dagegen die Fähigkeit, mit dem Benutzer zu kommunizieren. Hier ist es offenbar nicht möglich, den Benutzer zu replizieren, der an einem Punkt mit dem System verbunden ist. Stattdessen müssen andere Methoden eingesetzt werden, um Zuverlässigkeit zu gewährleisten. Dem gegenüber kann ein Dienst zum Speichern von Dateien diese Dateien nach Belieben replizieren, sodass sowohl Erreichbarkeit als auch Zuverlässigkeit durch Replikation erreicht werden können.

Ein Beispiel: Ein P2P-Speichersystem kann seine Zuverlässigkeit dadurch steigern, dass es Dateien oder Blöcke mit vielen Peers repliziert. Solange die Replikate richtig verteilt sind, können die Dateien dadurch immer noch zuverlässig bereitgestellt werden, selbst wenn einige Peers das System verlassen sollten.

- *Robustheit* (Robustness) ist die Fähigkeit eines Systems, den Betrieb während der gewöhnlichen Dynamik des Systems oder während beliebiger Systemstörungen aufrechtzuerhalten. Unter der gewöhnlichen Dynamik verstehen wir dabei das übliche Verhalten von voneinander unabhängigen Peers, die nach Belieben am System teilnehmen und es wieder verlassen. Um Robustheit möglich zu machen, muss ein Peer, der das System verlässt, sicherstellen, dass ein anderer Peer dessen Aufgaben übernimmt. Dabei könnten etwa Dateien oder Metadaten auf den anderen Peer kopiert werden. Darüber hinaus ist Fehlertoleranz ein wesentlicher weiterer Teil von Robustheit. Fehlertoleranz setzt voraus, dass ein System auch unvorhergesehene, zufällige Störungen von Peers überlebt, oder auch unvorhergesehene Trennungen einzelner Peers. In diesen Fällen sind zusätzliche Recovery-Mechanismen nötig, um einen ordnungsgemäßen Zustand des Systems wiederherzustellen. Derartige Mechanismen sollen dabei die Auswirkungen von Systemstörungen so gering wie möglich halten. Genauso wie Verfügbarkeit und Zuverlässigkeit kann auch die Robustheit eines Systems durch Replikation gesteigert werden. Robustheit und Fehlertoleranz sind zum Teil in der Zuverlässigkeit enthalten. Es ist unwahrscheinlich, dass ein System, das nicht mit Fehlern umgehen kann, zuverlässig funktioniert. Robustheit bezieht sich allerdings auf die Aktionen jedes einzelnen Peers, wohingegen sich Verfügbarkeit und Zuverlässigkeit auf die vom System angebotenen Dienste beziehen. In diesem Sinne ist Robustheit ein grundlegenderes Konzept und wird von der Verfügbarkeit und der Zuverlässigkeit benötigt.

Als Beispiel erhöht eine DHT ihre Robustheit dadurch, dass Peers die auf ihnen gespeicherten Daten an deren Nachbarn weitergeben, sollten sie das System verlassen. Dies reicht für die gewöhnliche Dynamik des Systems aus, während dessen es nicht zu unvorhergesehenen Störungen kommt. Um auch Störungen handhaben zu können, wird die Information auf einigen anderen Peers repliziert, ausserdem überprüft jeder Peer regelmässig, ob seine Nachbarn erreichbar sind.

Herkömmlicherweise werden die Qualitätsmerkmale Verlässlichkeit und Sicherheit unabhängig voneinander mit eigenen Modellen (z.B. Rollenbasierte-Zugriffskontrollmodelle in der Sicherheit und Modelle für fehlertolerante Systeme im Bereich der Verlässlichkeit) und eigenen Mechanismen (z.B. Verschlüsselung und Zugriffskontrolllisten im Bereich Sicherheit und Replikate im Bereich Verlässlichkeit) erforscht. Zwischen den Merkmalen der Sicherheit und Verlässlichkeit bestehen jedoch sehr enge Wechselwirkungen. Um beispielsweise die Verfügbarkeit und Robustheit eines Systems zu erhöhen, werden normalerweise Daten repliziert verwaltet. Unterliegen diese Daten hohen Vertraulichkeitsanforderungen, so werden sie verschlüsselt abgelegt und die Weitergabe ist streng reglementiert. Die zur Replikaterzeugung, -verteilung und -verwaltung eingesetzten Verfahren berücksichtigen jedoch derartige Sicherheitsanforderungen und eingesetzte Mechanismen nicht, so dass es zu Konflikten kommen kann, da z.B. Replikate aus Performanz- oder Lokalitätsgründen auf bestimmten Knoten erzeugt werden, die aber ggf. nicht vertrauenswürdig genug sind, um die sensible Information zu verwalten. Um die Informationsweitergabe zu kontrollieren, muss man also in einem solchen Fall zum einen die Vertraulichkeitsanforderungen bei der Replikaterzeugung berücksichtigen und zum anderen die Replikate auf den unterschiedlichen Peers vertrauenswürdig verwalten. Dazu sind die replizierten Daten insbesondere so zu verschlüsseln, dass eine konsistente Datenhaltung trotz Verschlüsselung möglich ist. Dies hat Einfluss auf die eingesetzten Sicherheitsmechanismen und -protokolle, die derartige Anforderungen bislang nicht berücksichtigen. Das Beispiel verdeutlicht, dass zur Erhöhung der Vertrauenswürdigkeit von P2P-Systemen die Untermerkmale Verlässlichkeit und Sicherheit eng zusammen erforscht werden müssen, um nachhaltige, qualitative Fortschritte zu erzielen.

4 P2P Mechanismen

4.1 Vorgehensweise

Ein P2P-System besteht aus Mechanismen, mit denen die Funktionalität des Systems erbracht wird. Zur Verbesserung und Kontrolle der Qualität eines P2P-Systems sind neue (zusätzliche) Mechanismen erforderlich oder die Modifizierung existierender.

Ein Mechanismus wird durch eine Anzahl von (konstruktiven) Regeln beschrieben, die die jeweils von den beteiligten Akteuren - typischerweise die Knoten des P2P-Netzwerks - durchzuführenden Aktionen beschreibt.

Beispiel: Im Zusammenhang mit dem Qualitätsmerkmal „Effizienz“ werden z.B. Mechanismen zur Optimierung des Routing (Wegefindung) im Overlay-Netzwerk oder Mechanismen an den Knoten zur Optimierung des Scheduling-Verhaltens der verschiedenen Verwaltungsnachrichten untersucht. Eine einfache mögliche Regel zur Optimierung des Routing-Mechanismus könnte folgende sein: Jeder Knoten erhöht den Hop-Zähler einer eingetroffenen Nachricht um 1. Überschreitet der Zähler nicht einen festgelegten Grenzwert *H*, so leitet der Knoten die eintreffende Nachricht an eine bestimmte Anzahl *n* zufällig ausgewählter direkt verbundener anderer Knoten weiter (Random Walk). Geeignete Werte für *H* und *n* wären zu erforschen.

Bei der Erforschung eines Mechanismus wird der in Abbildung 3 beschriebene iterative Prozess durchlaufen, der einen regelmäßigen Perspektivenwechsel zwischen einem deskriptiven und konstruktiven Blickwinkel beinhaltet. Aus einer gegebenen Qualitätsanforderung bezüglich eines oder mehrerer Qualitätsmerkmale werden allgemeine Constraints abgeleitet. Um sie zu erfüllen, werden Regeln formuliert. Diese Regeln werden erneut zusätzliche bzw. veränderte Constraints bedingen, die dann im nächsten Iterationsschritt in zusätzlichen bzw. geänderten Regeln münden. Der gesamte Mechanismus wird somit zunehmend detaillierter.

Beispiel: Um eine hohe Zuverlässigkeit in einem P2P-System zu erreichen, könnte ein Constraint beispielsweise lauten „*Jede im P2P-System gehaltene Datei sol zu jeder Zeit mit 99,9% Wahrscheinlichkeit verfügbar sein*“. Dieser Constraint könnte man durch die Regel: „*Dateien werden im System mindestens n-mal repliziert gehalten*“ versuchen zu realisieren. Diese Regel führt zu weiteren Constraints wie z.B. „*Die Anzahl vorhandener Replika muss im P2P-System gemessen werden können*“, „*Ein Knoten muss Dateien auf andere Knoten replizieren können*“, „*Das System muss die Anzahl n von nötigen Replikas abschätzen können*“. Diese führen dann zu weiteren Regeln, mit denen diese Constraints erfüllt werden. Eine Regel für den letztgenannten Constraint könnte die Anzahl *n* dezentral aus der Ausfallwahrscheinlichkeit der Knoten berechnen; dies führt wiederum zu dem Constraint „*Ein Knoten muss die Ausfallwahrscheinlichkeit seiner Nachbarn abschätzen können*“. Letzteres läßt sich über die Regeln „*Über die Online-Zeiten der verschiedenen Nachbarknoten wird Buch geführt*“ und „*Aus den gemessenen Online-Zeiten wird die Ausfallwahrscheinlichkeit eines Knotens berechnet*“ realisieren.

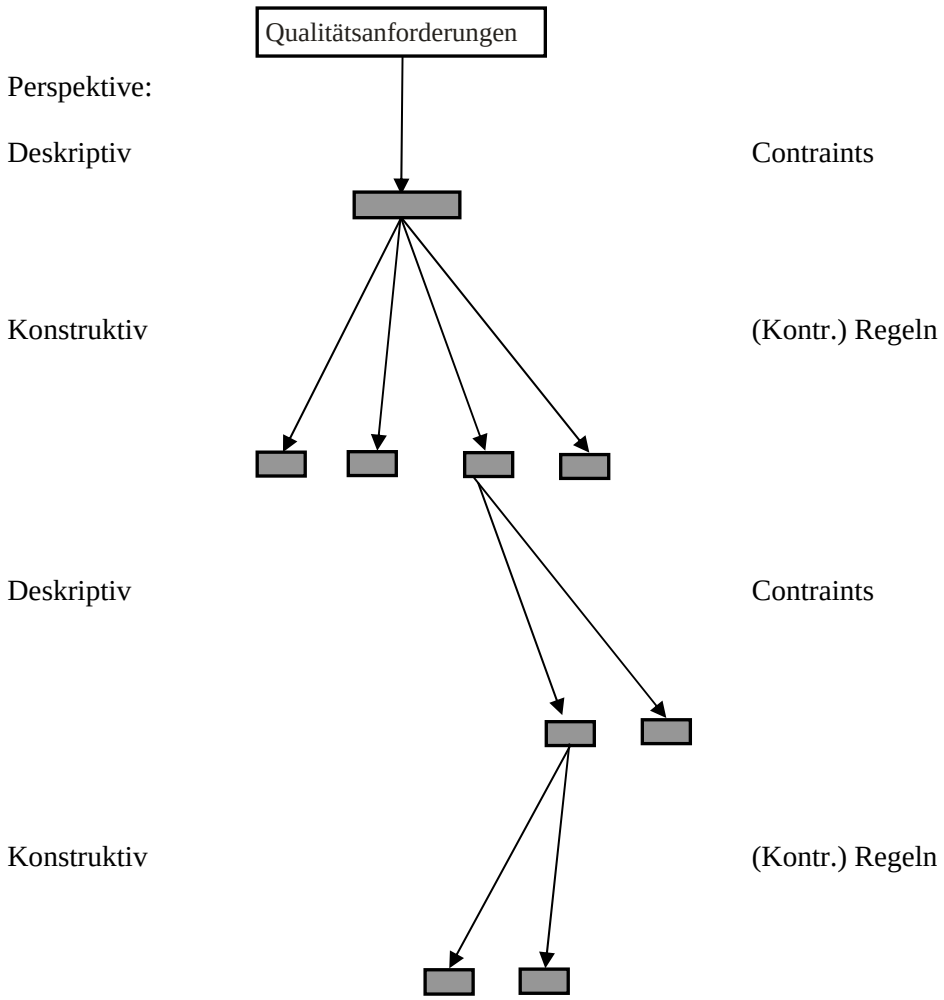

Abbildung 3: Vorgehensweise bei der Erforschung eines Mechanismus

4.2 Durchsetzung der Regeln

Eine wesentliche Eigenschaft, in der sich das Peer-to-Peer Paradigma von anderen Kommunikationsparadigmen unterscheidet, ist die Autonomie der Teilnehmer. Auf Grund dessen ist in Peer-to-Peer-Systemen mit eigennützigem oder böswilligem Verhalten der Teilnehmer zu rechnen, d.h. einige Teilnehmer werden sich eigennützig oder böswillig nicht an die vorgegebenen Regeln halten. Bei einem Client-Server-System kann der Server die Einhaltung der Regeln erzwingen, indem er im einfachsten Fall bei Nichteinhaltung die Diensterbringung verweigert. Ein P2P-System sollte sicherstellen können, dass die Mechanismen von den Knoten korrekt umgesetzt werden. Dies kann zum einen erreicht werden, indem die Regeln der Mechanismen so entworfen werden, dass kein Knoten einen Anreiz hat (im spieltheoretischen Sinn), sich nicht an sie zu halten. Ein Beispiel hierfür wäre eine Vickrey Auktion [LR00], bei der ein Knoten sich direkt selbst schadet, wenn er nicht seine wahren Präferenzen o_enbart. In den meisten Fällen wird eine solche Lösungsmöglichkeit allerdings nicht zur Verfügung stehen, so dass in diesen Fällen mit zusätzlichen Regeln die Mechanismen/Regeln durchgesetzt werden sollten. Hierfür existieren zwei grundlegenden Strategien (sowie Varianten und Kombinationen davon):

1. *Anreize*

Bei dieser Strategie werden den Knoten positive Anreize („Belohnungen“) geboten, wenn sie sich an die Regeln halten. Der positive Anreiz kann beispielsweise in einer Verbesserung der allgemeinen Dienstqualität bestehen.

Beispiel: Ein Anreizverfahren für obiges Beispiel der Optimierung des Wegeleitverfahrens ist die Verwendung eines „Bezahl“-systems, bei dem ein Knoten für die Weiterleitung einer Nachricht ein Token in einer virtuellen Währung erhält. Mit diesem Token kann er dann Dienste anderer Knoten benutzen, siehe z.B. [McC01].

2. Strafe

Bei dieser Strategie werden Knoten, die einen Regelverstoß begehen, von den anderen Knoten „bestraft“. Die Strafe kann beispielsweise in einer Verschlechterung der allgemeinen Dienstqualität bestehen.

Beispiel: Im obigen Beispiel der Optimierung des Wegeleitverfahrens könnten Nachrichten von diesen Knoten absichtlich nicht weitergeleitet werden oder die Verbindung zu solchen Knoten getrennt werden. Man erkennt anhand dieser Problematik, dass das P2P-Paradigma neuartige Herausforderungen und Forschungsaspekte mit sich bringt, die bislang bei traditionellen Client-Server-Systemen nur eine geringe Rolle spielten.

4.3 Zielkonflikte und Abhängigkeiten

Zwischen den verschiedenen Qualitätsmerkmalen bestehen gegenseitige Abhängigkeiten und Zielkonflikte.

Beispiele: Eine Optimierung des Wegeleitverfahrens im Hinblick auf den Verwaltungsoverhead- wie im Beispiel für das Qualitätsmerkmal Effizenz oben geschildert - kann zu einer Verschlechterung der Verlässlichkeit und Lokalisierbarkeit führen, da Suchanfragen eventuell weniger Knoten erreichen als beim Fluten. Eine Erhöhung der Verlässlichkeit mit der Hilfe z.B. von Datenreplikation kann zu einer Verschlechterung der Adaptivität, Effizienz und Sicherheit führen.

Die Zielkonflikte haben dazu geführt, dass heute im Allgemeinen zwar gesamte P2P-Systeme erforscht und entwickelt werden, jedoch die Qualitätsmerkmale nur (wenn überhaupt) aus Analysesicht betrachtet werden. Der umgekehrte Schritt einer Synthese beruhend u.a. auf dem grundsätzlichen Verständnis dieser Zielkonflikte wird von uns im Projekt QuaP2P angestrebt.

5 Fazit

In diesem Technischen Report haben wir eine pragmatische Definition der Qualität von Peer-to-Peer-Systemen aus wissenschaftlicher Sicht gegeben.

Dazu haben wir eine Taxonomie von Qualitätsmerkmalen und Untermerkmalen aufgestellt. Im Rahmen der Forschergruppe QuaP2P (www.quap2p.de) werden diese Merkmale wissenschaftlich untersucht mit dem Ziel, die Qualität von P2P-Systemen zu verbessern.

Literaturverzeichnis

[AAA+03] I. Abraham, B. Awerbuch, Y. Azar, Y. Bartal, D. Malkhi und E. Pavlov: A Generic Scheme for Building Overlay Networks in Adversarial Scenarios. In: Proceedings of International Parallel and Distributed Processing Symposium (IPDPS'03), Seite 40, 2003.

[Ada99] L. A. Adamic: The Small World Web. In: Proceedings of the 3rd European Conference on Research and Advanced Technology for Digital Libraries (ECDL), Seiten 443-452, 1999.

[AG03b] N. B. Azzouna und F. Guillemin: Analysis of ADSL Traffic on an IP Backbone Link. In: Proceedings of the IEEE Global Communications Conference (Globecom 2003), Seiten 3742-3746, 2003.

[ALPH01] L. A. Adamic, R. M. Lukose, A. R. Puniyani und B. A. Huberman: Search in power-law networks. Physical Review E, 64(046135), 2001.

[And01] D. Anderson: Peer-to-Peer - Harnessing the Power of Disruptive Technologies, Kapitel SETIhome, Seiten 45-50. O'Reilly, 2001.

[AS03] J. Aspnes und G. Shah: Skip Graphs. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Seiten 384-393, 2003.

[BBC+98] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang und W. Weiss: An Architecture for Differentiated Services. RFC 2475, 1998.

[BBK+78] B. W. Boehm, J. R. Brown, H. Kaspar, M. Lipow, G. J. MacLeod und M. J. Merritt: Characteristics of Software Quality. North-Holland Publishing Company, New York, NY, 1978.

[BCS94] R. Braden, D. Clark und S. Shenker: Integrated Services in the Internet Architecture: an Overview. RFC 1633, 1994.

[BKK+03] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris und I. Stoica: Looking up data in P2P systems. Communications of the ACM, 46(2):43-48, 2003.

[BKLW95] M. Barbacci, M. H. Klein, T. H. Longsta und C. B. Weinstock: Quality Attributes. Technischer Bericht CMU/SEI-95-TR-021, Carnegie Mellon University, Software Engineering Institute, 1995.

[Blo70] B. H. Bloom: Space/time trade-offs in hash coding with allowable errors. Communications of the ACM, 13(7):422-426, 1970.

[Bon02] E. Bonsma: Fully Decentralized, Scalable Look-up in a Network of Peers using Small World Networks. In: Proceedings of Systemics, Cybernetics and Informatics (SCI), 2002.

[BR01] A. Basu und J. Riecke: Stability Issues in OSPF Routing. In: Proceedings of ACM SIGCOMM 2001, Seiten 225-236, 2001.

[BZB+97] B. Braden, L. Zhang, S. Berson, S. Herzog und S. Jamin: Resource ReServation Protocol (RSVP) - Version 1 Functional Specification. RFC 2205,1997.

[Cli00] Clip2: The Gnutella Protocol Specification v0.4n, 2000. http://www9.limewire.com/ developer/ gnutella_protocol_0.4.pdf.

[Com99] A. F. T. Commitee: Traffic Management (TM) Specification 4.1, 1999.

[CSWH00] I. Clarke, O. Sandberg, B. Wiley und T. W. Hong: Freenet: A Distributed Anonymous Information Storage and Retrieval System. In: ICSI Workshop on Design Issues in Anonymity and Unobservability, Seiten 46-66, 2000.

[dB46] N. de Bruijn: A combinatorial problem. In: Proceedings of the Koninklije Nederlandse Academie van Wetenshapen, Seiten 758-764, 1946.

[Din] DIN 66272 Bewerten von Softwareprodukten - Qualitätsmerkmale und Leitfaden zu ihrer Verwendung.

[DMS04] V. Darlagiannis, A. Mauthe und R. Steinmetz: Overlay Design Mechanisms for Heterogeneous, Large Scale, Dynamic P2P Systems. Journal of Networks and System Management, 12:371-395, 2004.

[DMS05] V. Darlagiannis, A. Mauthe und R. Steinmetz: Optimizing Overlay Network Stability using Burn-In Methods. Submitted for publication, 2005.

[DW88] M. S. Deutsch und R. R. Willis: Software Quality Engineering: A Total Technical and Management Approach. Prentice-Hall, Englewood Cliffs, NJ, USA, 1988.

[Fas] FastTrack. http://www.fasttrack.nu.

[FG03] P. Fraigniaud und P. Gauron: An Overview of the Content-Addressable Network D2B. In: Annual ACM Symposium on Principles of Distributed Computing, Seite 151, 2003.

[GBL+03] I. Gupta, K. Birman, P. Linga, A. Demers und R. van Renesse: Kelips: Building an Efficient and Stable P2P DHT Through Increased Memory and Background Overhead. In: Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS03), Seiten 160-169, 2003.

[GSGM03] P. Ganesan, Q. Sun und H. Garcia-Molina: YAPPERS: A Peer-to-Peer Lookup Service Over Arbitrary Topology. In: Proceedings of IEEE Infocom, 2003.

[HBMS04] O. Heckmann, A. Bock, A. Mauthe und R. Steinmetz: The eDonkey File-Sharing Network. In: Proceedings of the Workshop on Algorithms and Protocols for Efficient Peer-to-Peer Applications, GI Jahrestagung, Seiten 224-228, 2004.

[Hei03] Heise Verlag, c't Magazin: Nervige Port-Scans. http://www.heise.de/ct/faq/qna/nervige-port-scans.shtml, 2003.

[HJS+03] N. J. Harvey, M. B. Jones, S. Saroiu, M. Theimer und A. Wolman: SkipNet: A Scalable Overlay Network with Practical Locality Properties. In: Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems (USITS '03), Seiten 113-126, 2003.

[HK03] K. Hildrum und J. Kubiatowicz: Asymptotically Efficient Approaches to Fault-Tolerance in Peer-to-Peer Networks. In: Proceedings of 17th International Symposium on Distributed Computing, Seiten 321-36, 2003.

[HKRZ02] K. Hildrum, J. D. Kubiatowicz, S. Rao und B. Y. Zhao: Distributed object location in a dynamic network. In: Proceedings of the fourteenth annual ACM symposium on Parallel algorithms and architectures, Seiten 41-52, 2002.

[HLD+05] O. Heckmann, N. Liebau, V. Darlagiannis, A. Bock, A. Mauthe und R. Steinmetz: From Integrated Publication and Information Systems to Information and Knowledge Environments: Essays Dedicated to Erich J. Neuhold on the Occasion of His 65th Birthday, Band 3379 der Reihe Lecture Notes in Computer Science, Kapitel A Peer-to-Peer Content Distribution Network, Seiten 69-78. Springer-Verlag GmbH, 2005.

[HSL+06] R. Steinmetz, O. Heckmann, N. Liebau, A. Buchmann, C. Eckert, J. Kangasharju, M. Mühlhäuser, A. Schürr: Referenzszenarien für P2P Systeme: Technical Report KOM-TR-2006-04, TU Darmstadt - Multimedia Communications Lab, Mai 2006.

[ISO] ISO/IEC 9126: Software Engineering - Product Quality.

[Kal03] S. L. S. W. Kalepu, Sravanthi; Krishnaswamy: Verity: A QoS Metric for Selecting Web Services and Providers. In: Proceedings of the 4th International Conference on Web Information Systems Engineering Workshops (WISEW 2003), Seiten 131-139, 2003.

[KBC+00] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, C. Wells und B. Zhao: OceanStore: an architecture for global-scale persistent storage. In: Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, Seiten 190-201, 2000.

[Kil01] F. Kileng: Peer-to-Peer File Sharing Technologies - Napster, Gnutella and Beyond. R&D Report 18/2001, Telenor, 2001. http://www.telenor.no/ fou/ publisering/ Rapp01/ R18_2001.PDF.

[KK03] F. Kaashoek und D. R. Karger: Koorde: A Simple Degree-optimal Hash Table. In: Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS03), Seiten 98-107, 2003.

[KLL+97] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine und D. Lewin: Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web. In: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing, Seiten 654-663, 1997.

[LCC+02] Q. Lv, P. Cao, E. Cohen, K. Li und S. Shenker: Search and replication in unstructured peer-to-peer networks. In: Proceedings of the 16th International Conference on Supercomputing, Seiten 84-95, 2002.

[LKR04] J. Liang, R. Kumar und K. Ross: The KaZaA Overlay: A Measurement Study. Technischer Bericht, Polytechnic University, New York, 2004.

[LN97] W. Litwin und M.-A. Neimat: LH*S: A High-Availability and High-Security Scalable Distributed Data Structure. In: Proceedings of Research Issues in Data Engineering (RIDE), Seiten 141-150, 1997.

[LNS96] W. Litwin, M.-A. Neimat und D. A. Schneider: LH*S: a scalable, distributed data structure. ACM Transactions on Database Systems (TODS), 21(4):480-525, 1996.

[LR00] D. Lucking-Reiley: Vickrey Auctions in Practice: From Nineteenth-Century Philately to Twenty-First-Century E-Commerce. Journal of Economic Perspectives, 14(3):pp. 183-192, 2000.

[LRS02] Q. Lv, S. Ratnasamy und S. Shenker: Can Heterogeneity Make Gnutella Scalable? In: Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS02), Seiten 94-103, 2002.

[MBR03] G. S. Manku, M. Bawa und P. Raghavan: Symphony: Distributed Hashing in a Small World. In: Proceedings of the 4th USENIX Symposium on Internet Technologies and System (USITS '03), Seiten 127-140, 2003.

[McC01] J. McCoy: Mojo Nation Responds, 2001. http://www.openp2p.com/ pub/a/ p2p/ 2001/ 01/ 11/ mojo.html.

[MCR03] R. Mahajan, M. Castro und A. Rowstron: Controlling the Cost of Reliability in Peer-to-Peer Overlays. In: Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS03), Seiten 21-32, 2003.

[Mil67] S. Milgram: The small world problem. Psychology Today, 1:60-67, 1967.

[MM02] P. Maymounkov und D. Maziéres: Kademlia: A Peer-to-peer Information System Based on the XOR metric. In: Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS02), Seiten 53-65, 2002.

[MNR02] D. Malkhi, M. Naor und D. Ratajczak: Viceroy: a scalable and dynamic emulation of the butterfly. In: Proceedings of the 21th Annual Symposium on Principles of Distributed Computing, Seiten 183-192, 2002.

[MS03] J. Mischke und B. Stiller: Peer-to-Peer Overlay Network Management Through AGILE. In: Kluwer Academic Publishers, IFIP/IEEE International Symposium on Integrated Network Management (IM), Seiten 337-350, 2003.

[MWE87] J. M. Michael W. Evans: Software Quality Assurance and Management. John Wiley & Sons, Inc., New York, NY, USA, 1987.

[Neu94] B. C. Neuman: Readings in Distributed Computing Systems, Kapitel Scale in Distributed Systems, Seiten 463-489. IEEE Computer Society, 1994.

[NW03b] M. Naor und U. Wieder: A Simple Fault Tolerant Distributed Hash Table. In: Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS03), Seiten 88-97, 2003.

[Pug90] W. Pugh: Skip lists: a probabilistic alternative to balanced trees. Communications of the ACM, 33(6):668-676, 1990.

[Qua] DIN EN ISO 8402 - Quality management and quality assurance -Vocabulary.

[RD01] A. Rowstron und P. Druschel: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Seiten 329-350, 2001.

[RFH+01] S. Ratnasamy, P. Francis, M. Handley, R. Karp und S. Schenker: A scalable content-addressable network. In: Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Seiten 161-172, 2001.

[RK02] S. Rhea und J. Kubiatowicz: Probabilistic location and routing. In: Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), 2002.

[San03] Sandvine Incorporated: Regional Characteristics of P2P - File Sharing as a Multi-Application, Multi-National Phenomenon. White Paper, 2003.

[Sch01a] J. Schmitt: Heterogeneous Network Quality of Service Systems. Kluwer Academic Publishers, 2001.

[Shi01] C. Shirky: Peer-to-Peer - Harnessing the Power of Disruptive Technologies, Kapitel Listening to Napster, Seiten 19-28. O'Reilly, 2001.

[SMLN 03] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. F. Kaashoek, F. Dabek und H. Balakrishnan: Chord: A scalable Peer-to-Peer Lookup Service for Internet Applications. IEEE Transactions on Networking, 11(1):17-32, 2003.

[SW04b] R. Steinmetz und K. Wehrle: Peer-to-Peer-Networking & -Computing. Informatik Spektrum, Aktuelles Schlagwort, 27(1):51-54, 2004.

[Tut04] K. Tutschku: A Measurement-Based Traffic Profile of the eDonkey Filesharing Service. In: Proceedings of the 5th Anual Passive & Active Measurement Workshop, Seiten 12-21, 2004.

[vM01] A. van Moorsel: Metrics for the Internet Age: Quality of Experience and Quality of Business. Technischer Bericht HPL-2001-179, HP Labs, 2001. Also published in 5th Performability Workshop, September 2001, Erlangen, Germany.

[YGM02] B. Yang und H. Garcia-Molina: Improving Search in Peer-to-Peer Networks. In: 22nd International Conference on Distributed Computing Systems (ICDCS' 02), Seiten 5-14, 2002.

[Wik] Wikipedia, Definition von „Effizienz". http://de.wikipedia.org/wiki/Effizienz.

[WXZ03] S. Wang, D. Xuan und W. Zhao: On Resilience of Structured Peer-to-Peer Systems. In: Proceedings of IEEE Global Telecommunications Conference (GLOBECOM), 2003.

[ZGG02] H. Zhang, A. Goel und R. Govindan: Using the Small-World Model to Improve Freenet's Performance. In: Proceedings of IEEE Infocom, Seiten 79-79, 2002.

[ZHS+04] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph und J. Kubiatowicz: Tapestry: A Resilient Global-scale Overlay for Service Deployment. IEEE Journal on Selected Areas in Communications, 22(1):41-53, 2004.