

Problemseminar im WS 03/04 - Prof. Rahm

Peer-to-Peer (P2P) Data Management

# Einführung

Bearbeiter: Sarah Gebhardt

Betreuer: Ulrike Greiner

---

## Inhaltsverzeichnis

<b>1. Geschichte der P2P-Netze.....</b>	<b>3</b>
1.1 Beginn der Netzwerkentwicklung.....	3
1.2 Umsetzung der entwickelten Architektur.....	3
1.3 Entwicklung zu dem auf der Client/Server Architektur basierenden Internet.....	4
1.4 Wiederaufnahme des Peer-to-Peer Gedanken.....	4
<b>2. Definition und Modelle des P2P.....</b>	<b>5</b>
2.1 Was bedeutet Peer-to-Peer?.....	5
2.2 Eigenschaften von P2P-Netzwerken.....	6
2.3 Vergleich des P2P mit der Client/Server Architektur.....	7
2.4 Modelle des P2P.....	8
2.4.1 Pure P2P-Netzwerke.....	8
2.4.2 Hybride P2P-Netzwerke.....	9
2.4.3 Super-Peer Netzwerke.....	11
<b>3. Anwendungsgebiete.....</b>	<b>12</b>
3.1 Distributed Computing.....	12
3.1.1 Die Funktionsweise des Distributed Computing.....	12
3.1.2 SETI@home.....	13
3.2 Collaboration.....	14
3.2.1 Herausforderungen und Probleme.....	15
3.2.2 Instant Messaging.....	15
3.2.3 Groupware.....	16
3.2.4 Groove.....	16
<b>4. Herausforderungen und Anforderungen an P2P-Systeme.....</b>	<b>17</b>
4.1 Technische Probleme und Herausforderungen.....	17
4.1.1 Heterogenität und Erweiterbarkeit.....	17
4.1.2 Verwaltung von Peers und Ressourcen im P2P-Netz.....	17
4.1.3 Vertrauen, Sicherheit und Autonomie.....	18
4.2 Verwaltung von Ressourcen mit Hilfe von Datenbanken.....	19
4.3 Juristische Probleme und Herausforderungen.....	21
<b>5. Die Zukunft von P2P und Zusammenfassung.....</b>	<b>22</b>
<b>Literaturverzeichnis.....</b>	<b>23</b>

## 1. Geschichte der P2P-Netze

Heute wird der Begriff Peer-to-Peer schnell ausschließlich mit verschiedenen File-Sharing Systemen verbunden und darauf reduziert. Nur wenige wissen, dass P2P viel mehr als diese begrenzte Aufgabenstellung umfasst und dass das P2P-Konzept und die zugrunde liegenden Technologien keinesfalls eine neue Erfindung der späten 90er Jahre sind, denn schon vor rund 30 Jahren arbeiteten viele Unternehmen an Architekturen, die man heute als Peer-to-Peer bezeichnen würde. Die folgende Ausarbeitung soll in das Konzept des P2P und das damit verbundene Data Management einführen, indem sie die Geschichte des P2P, die grundlegenden Merkmale und Modelle, Anwendungsbereiche und Herausforderungen an das P2P aufzeigt.

### 1.1 Beginn der Netzwerkentwicklung

Anfang der 60er Jahre, während der Zeit des Kalten Krieges, wurde das erste P2P-Netzwerk entwickelt. Die USA stellte sich damals die Frage, wie man die Kommunikation zwischen den Machthabern, Militärs, Behörden etc. während und nach einem möglichen Atomkrieg aufrecht erhalten könne. Selbst wenn große Teile des Kommunikationsnetzes zerstört würden, sollten die noch funktionsfähigen Teile des Netzes weiterhin untereinander kommunizieren können.

1964 wurden dann von der 'RAND Corporation' die ersten Lösungsansätze für dieses Problem veröffentlicht. Die Kernaussagen waren:

- o das Netzwerk soll keine verletzbare zentrale Steuerung (Autorität) erhalten und deshalb auch, wenn Teile davon zerstört werden, weiter funktionsfähig bleiben
- o jede noch so kleine Kommunikationseinheit soll unabhängig von allen anderen und völlig selbstständig funktionsfähig sein, kein Knotenpunkt innerhalb des Netzes soll gegenüber einem anderen ausgezeichnet sein [NS02]

Damit diese zwei Konzepte realisiert werden konnten, mussten alle Kommunikationsknoten den gleichen Status im Netzwerk haben. Jedem Knoten sollte es möglich und erlaubt sein, Daten zu erzeugen, zu verschicken und zu empfangen. Dabei wurden die Daten in Pakete zerteilt, die jeweils mit dem Empfänger und Absender versehen wurden. Diese Pakete wurden einzeln verschickt, wobei die Route eines Pakets durch das Netz zum Empfänger nicht vorgegeben wurde. Die Datenpakete sollten sich selbst einen möglichen Weg suchen, über den sie das Ziel erreichen konnten. Welche Route das Paket wählte, war dabei völlig unwichtig, solange es beim Empfänger ankam. Der Vorteil bestand darin, dass nicht die ganze Nachricht, sondern nur kleine Pakete wiederholt werden mussten, wenn sie ihr Ziel nicht erreichten. Außerdem konnten, selbst wenn große Teile des Netzes zerstört waren, Daten immer noch sicher den Empfänger erreichen, solange dieser nur irgendwie mit dem Netz verbunden war. Die Daten mussten dazu einfach nur eine noch funktionsfähige Route durchs Netz nehmen.

Dieses System war somit im höchsten Grade ausfallsicher, da Informationen im funktionierenden Teil immer weiter ausgetauscht werden konnten.

### 1.2 Umsetzung der entwickelten Architektur

Das erste Netzwerk, das auf diesen Grundlagen aufgebaut wurde, war 1968 ein kleines Teilnetz in Großbritannien. Kurz darauf folgte das Pentagon, das 1969 ein Netzwerk mit vier Knotenpunkten aufgebaut hatte. Das Netzwerk verband vier Universitäten und erhielt den Namen **ARPA-Net** nach der "Advanced Research Projects Agency", die federführend beim Aufbau des Netzes war. Dieses ARPA-

Net wird heute als die Mutter des Internets und Urform eines P2P-Netzwerks bezeichnet. [NS02] Die Herausforderung bestand darin, unterschiedliche Netzwerke und zukünftige Technologien in einem Netzwerk zu verbinden. Das Netz diente dann vor allem dem Austausch von Daten und der Kommunikation zwischen Forschern.

Das ARPA-Net entwickelte sich schnell weiter. Ende 1971 bestand das Netz bereits aus 15 Knoten und 1972 schon aus 37. In den folgenden Jahren wurde das Netzwerk immer größer, da durch die dezentrale Struktur schnell und problemlos neue Knoten angeschlossen werden konnten, wenn diese das TCP/IP (Transmission Control Protocol / Internet Protocol) verstanden. Diese TCP/IP Protokollfamilie wurde 1972 eingeführt, damit auch eine Datenübertragung zwischen verschiedenen Hardware-Plattformen, autonomen Netzen und Rechnern mit verschiedenen Betriebssystemen ermöglicht werden konnte. Damals war es selbstverständlich, dass jeder Knoten des Netzwerks nicht nur Dienste anderer Knoten in Anspruch nahm, sondern selber auch Dienste zur Verfügung stellte. Das Netz entwickelte sich also zu einem Kommunikationsmedium, das zum Austausch von Ressourcen zwischen Rechnern benutzt wurde, die im Netz gleichberechtigt waren und war damit ursprünglich ein P2P-System.

### 1.3 Entwicklung zu dem auf der Client/Server Architektur basierenden Internet

Im Laufe der 80er Jahre konnten immer mehr Gesellschaftsgruppen leistungsfähige Computer benutzen und kaufen. Durch die dezentrale Struktur des ARPA-Netzes konnten sich diese ungehindert und einfach in das Netz einbinden. Das Netz war also jetzt nicht mehr nur für Regierung, Universitäten oder Militärs, sondern auch für die breite Öffentlichkeit zugänglich. Dadurch entstand ein großes, weit aufgespanntes Netz aus vielen lokalen Netzwerken, das man als **Internet** bezeichnete. Im ursprünglichen Netzwerk sollten die Teilnehmer statische und permanente IP-Adressen haben und ständig am Netz angeschlossen sein. Die Adressierung wurde durch das Domain Name System (DNS) geregelt. Es war ungewöhnlich und selten, dass ein Rechner seine IP-Adresse änderte. Mit der Entwicklung des ersten Web-Browsers Mosaic 1993 wurde jedoch dieses Modell zurückgedrängt. Damit ein PC den Browser benutzen konnte, musste er sich nur über ein Modem mit einer eigenen IP-Adresse an das Netz anschließen. Da sich dadurch PCs beliebig an- und abmelden durften und man sowieso einen zukünftigen Mangel an IP-Adressen voraussah, wurden nun die IP-Adressen durch die ISPs (Internet Service Provider) dynamisch den Teilnehmern zugewiesen. Dadurch war es nicht mehr so einfach möglich auf diese Rechner zuzugreifen und konnte sie nicht als Server benutzen. Außerdem wurde Ende der 80er Jahre mit der Entwicklung der Ethernet-Verbindungstechnik die Client/Server Architektur in den Vordergrund gerückt. Diese konnte sehr viele Benutzer miteinander verbinden und die Verwaltung des Netzes war sehr billig, effizient und schnell, da alles über einen zentralen Server geregelt wurde. Die teilnehmenden Clients konnten einfache, billige und nicht besonders leistungsfähige Rechner sein. Durch diese Entwicklung veränderte sich auf einmal der ursprüngliche Gedanke von gleichberechtigten Knotenpunkten und eine große Anzahl von Teilnehmer verband sich mit dem Netz, die nicht mehr Dienste zur Verfügung stellen wollten. Sie agierten vielmehr allein als Konsumenten und nahmen Ressourcen in Anspruch, von nur noch verhältnismäßig wenigen Rechnern, die als Server ihre Dienste zur Verfügung stellten. Dies kann man auch darauf zurückführen, dass viele Rechner nur eine sehr langsame und einseitige Anbindung hatten und somit gar keine Ressourcen anbieten konnten. Dadurch entwickelte sich das Internet immer mehr zu einer reinen Client/Server Architektur und der P2P-Gedanke wurde vorläufig in den Hintergrund geschoben.

### 1.4 Wiederaufnahme des P2P-Gedanken

Während der 90er Jahre stieg die Anzahl der privaten Rechner und Rechnerarbeitsplätze in

Unternehmen sehr stark an, außerdem wuchs die Leistungsfähigkeit, Zuverlässigkeit und das Speicherplatzvolumen der PCs. Damit waren diese nun viel leistungsfähiger als Computer, die früher wegen ihres Leistungsvermögens als Server eingesetzt wurden. Auch die Technologie des Internets wurde weiter entwickelt und die Bandbreite erhöhte sich. Als die ersten Instant-Messaging Systeme eingeführt wurden, entwickelten sie sich schnell zu dem beliebtesten Kommunikationswerkzeug neben E-mails. Die Netzteilnehmer suchten bald nach weiteren Wegen direkt mit anderen über das Internet zu kommunizieren. Als die Leistungsfähigkeit und Geschwindigkeit der Clients und des Netzwerks weiter wuchsen, wurden zunehmend Musik, Bilder und andere Dateien ausgetauscht. Durch all diese Entwicklungen wurde der Weg zum P2P wieder geebnet. Programme wie Napster, ICQ, KaZaA etc. konnten sich schnell durchsetzen, fanden bald viele Nachahmer und wurden zur Grundlage der Entwicklung vieler P2P Technologien und Anwendungen.

Heute wird angenommen, dass die wieder entdeckte P2P-Technologie die Kommunikation, Interaktion und Zusammenarbeit sowohl auf privater als auch geschäftlicher Ebene grundlegend verändern wird. P2P bietet genau genommen die Chance, das Internet mit seiner eigentlichen zugrunde liegenden Architektur zu nutzen - als ein dynamisches weltweites Netzwerk aus Ressourcen, die alle direkt miteinander kommunizieren und interagieren können. [NS02]

## 2. Definition und Modelle des P2P

Von der P2P Community selbst werden verschiedene Definitionen verwendet:

Intel P2P working group:

The sharing of computer resources and services by direct exchange between systems. [KL+02]

Ross Lee Graham definiert P2P durch drei Anforderungen:

- they have an operational computer of server quality;
- they have an addressing system independent of DNS; and
- they are able to cope with variable connectivity [KL+02]

Clay Shirky von O'Reilly and Associate benutzt die folgende Definition:

P2P is a class of applications that takes advantage of resources storage, cycles, content, human presence available at the edges of the Internet. Because accessing these decentralized resources means operating in an environment of unstable connectivity and unpredictable IP addresses, P2P nodes must operate outside the DNS system and have significant or total autonomy from central servers [KL+02]

Im Folgenden soll zum Teil aufgrund der Definition von Clay Shirky der Grundgedanke des P2P näher erläutert werden und auf die Unterschiede zu den bisher vorherrschenden Client/Server Modellen eingegangen werden.

### 2.1 Was bedeutet Peer-to-Peer?

Der Begriff 'Peer' kommt aus dem Englischen und kann mit "Ebenbürtiger" übersetzt werden. Peer-to-Peer bedeutet also "Ebenbürtiger zu Ebenbürtiger" und trifft damit den Kerngedanken des P2P, nämlich ein Konzept, das ein Netzwerk beschreibt, in dem alle angeschlossenen Teilnehmer (Peers) völlig gleichberechtigt sind, also gleiche Fähigkeiten und Verantwortlichkeiten haben und sich gegenseitig

Ressourcen wie Daten, Speicher, Programme, Rechenleistung etc. zugänglich machen, diese gemeinsam nutzen und austauschen können und dabei idealerweise auf eine zentrale Instanz verzichten.

## 2.2 Eigenschaften von P2P-Netzwerken

Ein P2P-Netzwerk sollte die folgenden Eigenschaften erfüllen:

- jeder Peer ist **gleichberechtigt** und kann im Netz **als Server und Client gleichzeitig** agieren, d.h. er kann sowohl Ressourcen zur Verfügung stellen als auch Ressourcen anderer Peers in Anspruch nehmen, jeder kann Daten senden, speichern und empfangen und Dienstleistungen anbieten oder benutzen
- alle **Peers behalten die Kontrolle über ihr lokales System**, bleiben also **autonom**, sie allein bestimmen, was wann und mit wem ausgetauscht wird und ob oder wann sie sich an einem Netz an- oder abmelden
- daraus folgt, dass Peers **nicht permanent** angeschlossen sein müssen und es **nicht notwendig** ist, dass ein Peer eine **konstante Adresse** aufweist
- die Kommunikation zwischen den Peers erfolgt **in Echtzeit** und **direkt untereinander**, Daten werden nicht über einen Server geleitet, der die Übertragung in irgendeiner Weise verzögert
- Peers sind im Idealfall **von keinem zentralen System (Server) abhängig**
- die **Ressourcen** werden allein **durch die Peers zur Verfügung gestellt**, deshalb sind die Ressourcen über das ganze Netz **dezentral** verteilt und nicht an zentralen Punkten gespeichert
- alle **'Aktionen'** finden an den **Endpunkten des Netzes** also bei den Endgeräten statt
- die bisher aufgeführten Punkte führen dazu, dass das ganze Netz einen **dynamischen Aufbau** hat und zu jeder Zeit eine **andere Konfiguration** aufweisen kann

Eine der wichtigsten Eigenschaften bzw. Ziele des P2P, die oben nicht genannt wurden, sind jedoch die Aufgabe der P2P-Anwendungen. Diese sollen die **freien Ressourcen** (Speicherplatz, Rechenleistung, Dateien etc.), die jeder Peer aufweisen kann und die einen zu großen Wert haben um vergeudet zu werden, in diesem heterogenen, dynamischen und unstabilen Netzwerk **auffinden und verwalten** können.

Auf manche dieser Eigenschaften wird später noch näher eingegangen werden und weitere Eigenschaften können aus den obigen hergeleitet werden. Z.B. ist es in einem P2P-Netz aufgrund des dynamischen Aufbaus fast unmöglich Quality of Service zu gewährleisten. Die Performanz, Speicherplatzkapazitäten und andere QoS-Aspekte können sich stets ändern. Trotzdem hat das Netz als Ganzes einen genügend hohen Grad an Zuverlässigkeit. Nicht jede P2P-Anwendung erfüllt alle diese Kriterien. Im nächsten Kapitel wird man sehen können, dass es verschiedene Modelle des P2P gibt, die z.B. eine zentrale Komponente erlauben.

Das P2P-Konzept lässt sich am Anfang einfacher erfassen, wenn man die P2P-Architektur mit der bisher eingesetzten Client/Server-Architektur vergleicht. Dies soll im nächsten Abschnitt erfolgen.

### 2.3 Vergleich des P2P mit der Client/Server Architektur

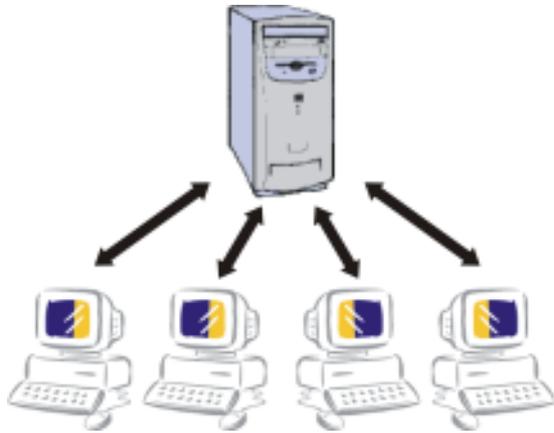


Abb. 1: Client/Server Architektur

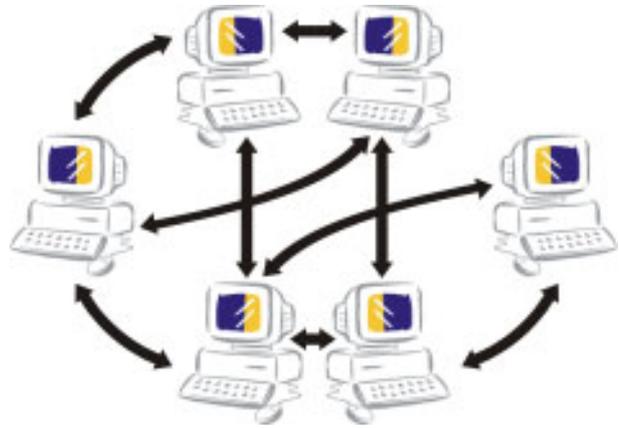


Abb. 2: P2P Architektur

<ul style="list-style-type: none"> <li>• jede Kommunikation und jeder Zugriff auf Ressourcen erfolgt über einen zentralen Server</li> </ul>	<ul style="list-style-type: none"> <li>• bei der Kommunikation sind die Peers nicht von einem zentralen Server abhängig</li> </ul>
<ul style="list-style-type: none"> <li>• der Server kann nur Anfragen entgegennehmen und keine absenden, Clients können nur Anfragen an den Server stellen jedoch keine entgegennehmen, Clients können untereinander nicht in Kontakt treten, beim Server allein liegt die gesamte Kontrolle</li> </ul>	<ul style="list-style-type: none"> <li>• Peers kommunizieren direkt untereinander und können eine direkte Verbindung zu jedem anderen Peer aufbauen um gegenseitig auf Ressourcen zuzugreifen</li> </ul>
<ul style="list-style-type: none"> <li>• Clients greifen nur auf den Server zu um einen Nutzen zu erhalten, steuern jedoch selbst nichts bei</li> </ul>	<ul style="list-style-type: none"> <li>• alle Peers sind gleichberechtigt und jeder Peer kann Ressourcen anbieten und benutzen</li> </ul>
<ul style="list-style-type: none"> <li>• es entsteht eine große Asymmetrie, es werden viel mehr Daten vom Server zum Client gesendet als andersherum</li> </ul>	<ul style="list-style-type: none"> <li>• es wird ein symmetrisches Protokoll benötigt, das nicht auf einer bestimmten Rollenverteilung basiert</li> </ul>
<ul style="list-style-type: none"> <li>• Ressourcen werden nur beim Server gespeichert, verwaltet und auf Anfrage an Clients übertragen</li> </ul>	<ul style="list-style-type: none"> <li>• Ressourcen werden nicht bei einem Server hinterlegt, sondern verlagern sich vom Zentrum weg an die Endpunkte des Netzes (den Peers), sind also dezentral auf viele Peers verteilt</li> </ul>
<ul style="list-style-type: none"> <li>• die Informationen liegen auf dem Server relativ statisch und sind abhängig von Aktualisierungen, die nur der Provider durchführt</li> </ul>	<ul style="list-style-type: none"> <li>• jeder Peer ist selbst verantwortlich für die Ressourcen, die er zur Verfügung stellt, und für deren Aktualität</li> </ul>
<ul style="list-style-type: none"> <li>• das gesamte Netz ist vom Server abhängig, fällt dieser aus, bricht die Funktionalität des Netzes zusammen</li> </ul>	<ul style="list-style-type: none"> <li>• es bleibt ohne Folgen, wenn sich Peers dynamisch an- und abmelden, das Netz kann ungehindert weiter funktionieren</li> </ul>

## 2.4 Modelle des P2P

Die P2P-Netzwerke werden nach ihrer Struktur unterteilt. Das ursprüngliche P2P-Netz kommt ohne zentrale Server aus, deshalb werden P2P-Netze, die keine zentrale Instanz haben, als

- **reine/pure** oder **unbrokered** (d.h. unvermittelte) P2P-Netzwerke

bezeichnet. Netze, die dennoch einen Server einsetzen um gewisse Aufgaben auszuführen, werden

- **hybride** (d.h. gemischte) oder **brokered** (d.h. vermittelte) P2P-Netzwerke

genannt.

### 2.4.1 Pure P2P-Netzwerke

#### Eigenschaften der puren P2P-Netze

Das reine P2P-Netz ist vollkommen dezentral organisiert. Das heißt, alle Teilnehmer im Netz sind gleichberechtigt und es gibt keinen, der spezielle kontrollierende oder administrative Aufgaben innehat. Es gibt also keine zentrale Stelle und keinen der besser gestellt ist als andere. Um die Verbindung zwischen den einzelnen Peers aufzubauen oder sie zu koordinieren, wird kein Server benötigt. Die Peers bilden zusammen die Administration. Jeder Knoten ist dabei gleichzeitig Client und Server, deshalb werden die Knoten auch manchmal als Servents bezeichnet (**Server** und **Client**). Da jeder Teilnehmer gleichzeitig und gleichberechtigt Daten und Ressourcen empfangen bzw. benutzen und weitergeben kann, werden die Daten über das ganze Netzwerk verteilt und sind auf den einzelnen Peers gespeichert. Im puren Modell sind deshalb Informationen sehr redundant bei vielen verschiedenen Teilnehmern gespeichert. P2P-Befürworter gehen deshalb sogar davon aus, dass die Redundanz dazu führt, dass in Zukunft für diese Informationen keine Backup-Systeme mehr nötig sind. [NS02]

#### Anmeldung und Suche im puren P2P-Netz

Meldet sich ein neuer Peer im Netzwerk an, sucht er nach einem anderen Peer, indem er z.B. auf eine Liste zugreift, auf der diejenigen Peers verzeichnet sind, die besonders häufig erreichbar sind, und eine Nachricht an einen dieser Peers schickt. Wird diese Nachricht von einem Teilnehmer im Netz empfangen, schickt dieser eine Antwort zurück und teilt dem neuen Peer gleichzeitig mit, zu welchen anderen Peers er eine Verbindung aufgebaut hat. Diesen teilt er wiederum mit, dass sich ein neuer Peer angemeldet hat. So können sich über das Schneeballprinzip in kürzester Zeit Tausende von Peers

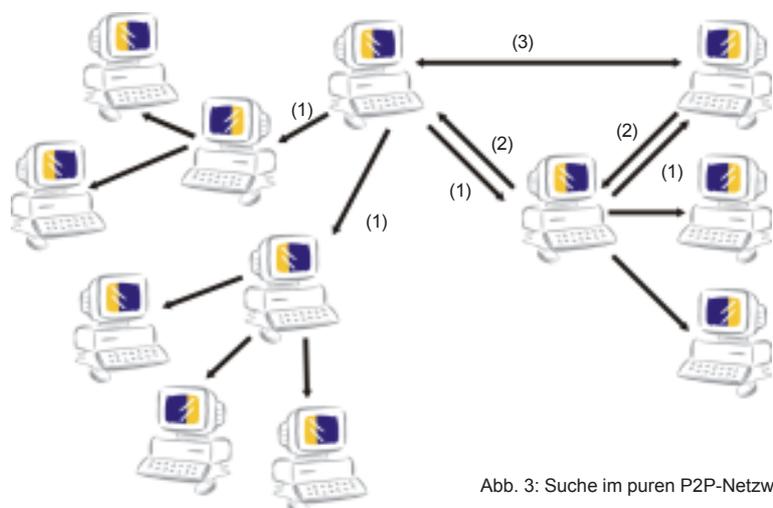


Abb. 3: Suche im puren P2P-Netzwerk

(1) Suchanfrage an alle verbundenen Peers (2) Antwort an suchenden Peer (3) direkter Ressourcenaustausch zwischen den Peers

verbinden. Damit diese Suche nicht unendlich fortgeführt wird oder Schleifen entstehen, wird der Suchanfrage ein Zähler zugeordnet ("Time-to-live"). Jeder Peer, der die Anfrage erhält, verringert diesen Zähler um eins und schickt erst dann die Nachricht an andere Peers weiter. Steht der Zähler auf 0, wird die Anfrage nicht weitergeschickt und die Suche ist damit beendet. So wird gewährleistet, dass die Effizienz des Netzwerks erhalten bleibt und die Suchanfrage nicht beliebig lange im Netz weitergegeben wird. Die Begrenzung, die damit erreicht wird, nennt man Horizont.

Die Suche nach Ressourcen beruht auf demselben Prinzip wie das Finden von anderen Peers. Der suchende Peer schickt die Suchanfrage an diejenigen Peers, mit denen er verbunden ist. Diese wiederum schicken sie an die Peers, mit denen sie eine Verbindung haben. Besitzt ein Peer die gewünschte Ressource, schickt er auf demselben Weg dem ursprünglich anfragenden Peer eine Antwort. Diese zwei Peers verbinden sich dann direkt um z.B. Dateien auszutauschen. (siehe Abb. 3, Seite 8)

### **Vorteile und Nachteile des puren P2P-Netzes**

Durch die dezentrale Struktur ist das Netzwerk an sich viel robuster als eine zentrale Client/Server Struktur. Das Netz kann nicht durch Serverausfälle lahm gelegt werden, da es von keiner zentralen Stelle abhängig ist. Auch das Netz schnell abzuschalten ist unmöglich. Da in diesem Modell Peers auf keinen Server zugreifen müssen, ist das Netz beliebig skalierbar, es gibt keinen zentralen Flaschenhals durch den die Performanz oder die Leistungsfähigkeit des Netzes gesenkt werden könnte.

Allerdings führt gerade diese Struktur auch zu mehreren Problemen und Herausforderungen. Schon beim Anmelden kann es zu Schwierigkeiten kommen, wenn man keinen anderen Peer findet, außerdem kann es durch das Schneeballprinzip zu Überlastungen des Netzes und damit zu Performanzeinbußen kommen. Auch die Suche in einem puren Netz ist deshalb langsamer, da die Suchanfragen erst durch das Netz geschickt werden müssen, was bei sehr vielen Anfragen und großen Netzen ohne geeignete Mechanismen schnell zu einem großen Verkehrsaufkommen führt, das wiederum das Netz belastet. Unter Umständen ist es möglich, dass Netze in Teilnetze ohne Verbindungen untereinander zerfallen, wenn sich einzelne Peers abmelden. Dazu kommen nicht-technische Probleme z.B. die fehlende Kontrolle oder Zensur von Daten, die einerseits gewollt und beabsichtigt ist, aber andererseits das Verbreiten von rechtswidrigen Inhalten unterstützt.

Die bekanntesten Beispiele für das pure P2P sind Gnutella und Freenet (beides File-Sharing Systeme, siehe P2P-Seminar, Thema 2).

## **2.4.2 Hybride P2P-Netzwerke**

### **Eigenschaften der hybriden P2P-Netze**

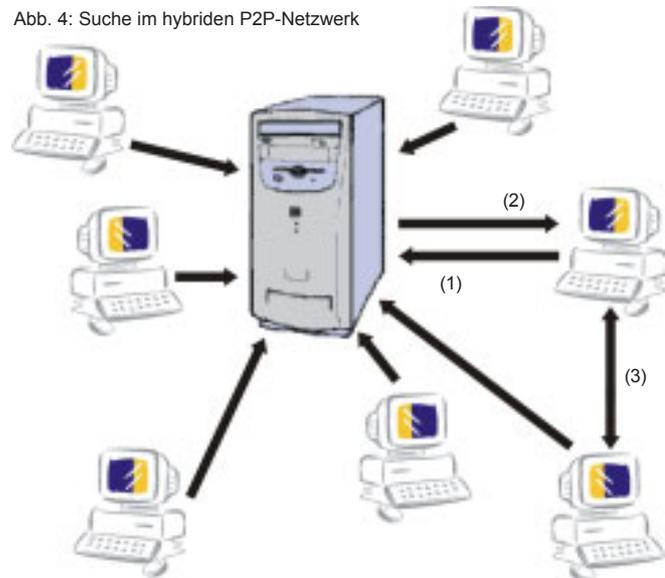
Im Gegensatz zu den puren P2P-Netzen setzen die hybriden Netze einen oder mehrere zentrale Server ein um die Vorteile der Client/Server Architektur zusätzlich zu den Vorteilen des P2P nutzen zu können. Durch die Server kann die Kommunikation zwischen den Peers und das Auffinden von Ressourcen vereinfacht und beschleunigt werden, da der Server eine Metadatenbank (Index) besitzt, die Informationen über die angemeldeten Peers, über die Dienste dieser Peers und die von ihnen angebotenen Ressourcen enthält.

## Anmeldung und Suche im hybriden P2P-Netz

Möchte sich ein neuer Peer mit dem P2P-Netz verbinden um Ressourcen auszutauschen, muss er sich nur mit dem zentralen Server in Verbindung setzen, meist werden dabei die auf dem Server gespeicherten Informationen über den Peer und seine Ressourcen aktualisiert. In dem Index dieses Servers kann der Peer nun direkt nach den Ressourcen, die andere Peers zur Verfügung stellen, suchen ohne auf diese Peers zuzugreifen oder das ganze Netz nach den gewünschten Inhalten durchsuchen zu müssen. Stellt ein Peer eine Anfrage an den Server, durchsucht dieser seine Datenbank und liefert dem Peer eine Liste von Dateien oder Diensten zurück, die mit der Suchanfrage übereinstimmen und die online verfügbar sind, da die Peers, die diese zur Verfügung stellen, gerade mit dem Netz verbunden sind. Erst wenn der Teilnehmer die Ressource nutzen, z.B. eine Datei herunterladen möchte, bekommt er vom Server die entsprechende Adresse des Peers, auf dem die Ressource hinterlegt ist. Dann kann er sich direkt mit diesem verbinden um die Daten von ihm herunterzuladen. Dabei erfolgt die Kommunikation nur noch zwischen den Peers.

Um über dieses System eine sehr große Anzahl von Nutzern verwalten zu können, braucht man nicht nur einen zentralen Server, sondern ein weit aufgespanntes Netz von Servern, das wiederum in Subnetzwerke aufgeteilt wird, die jeweils einen bestimmten Bereich des Netzes verwalten. Wenn sich ein neuer Peer über eine P2P-Anwendung anmeldet, wird dann automatisch nach dem Server gesucht, mit dem die beste Verbindung aufgebaut werden kann.

Abb. 4: Suche im hybriden P2P-Netzwerk



(1) Suchanfrage an den zentralen Server (2) Server sendet Antwort (3) direkter Ressourcenaustausch zwischen den Peers

## Vorteile und Nachteile des hybriden P2P-Netzes

Peers können in diesen hybriden Netzen sehr schnell und effizient Ressourcen finden und darauf zugreifen, da in den Servern ein zentraler Index geführt und ständig aktualisiert wird. Anders als bei den puren P2P-Netzen können hier die Ressourcen aller Netzteilnehmer gefunden und benutzt werden, da es keine Begrenzung gibt wie das TTL und alle Teilnehmer beim Server angemeldet sein müssen. Es müssen jedoch Lösungsmöglichkeiten dafür gefunden werden, falls Ressourcen nicht in dem kleineren Subnetzwerk, in dem der Peer angemeldet ist, gefunden werden. Es muss eine Möglichkeit geben auch in den anderen Teilen des Netzwerks, also auf anderen zentralen Servern, suchen zu können. Bei Napster wurde dieses Problem z.B. durch ein entsprechendes Plug-In gelöst, das eine netzweite Suche unterstützt. Es kann den Peers auch ermöglicht werden sich bei einem Server ihrer Wahl anzumelden. Auf diese Weise können auch Teilnetze entstehen, die nur ein bestimmtes Themengebiet umfassen und in denen man dann gezielt suchen kann.

Die Inhalte (Ressourcen) können bei den hybriden Netzen durch den Server kontrolliert und gefiltert werden, so dass z.B. illegale oder mit einem Urheberrecht versehene Daten nicht weiter verbreitet werden. Diese Kontrolle und Zensur kann natürlich auch als Nachteil gesehen werden.

Die Nachteile eines klassischen Client/Server Netzes, z.B. die Single-Point-of-Failure Problematik, können hier etwas abgeschwächt werden, da die Kommunikation zwischen den Peers und dem Server sehr gering ist und der eigentliche Datenaustausch zwischen den Peers stattfindet. Trotzdem

kann es auch hier zum Flaschenhals kommen, wenn das Netz sehr groß ist und viele Peers auf den Server zugreifen. Durch die Leistungsfähigkeit der zentralen Server wird außerdem die Skalierbarkeit und Performance des Netzes bestimmt. Durch einen Fehler im Server können ganze Teilnetze stillgelegt werden, wenn sich deren Teilnehmer nicht mit anderen Server verbinden können. Ein ganzes P2P-Netz kann abgeschaltet werden, indem man einfach die Server abschaltet.

Napster (File-Sharing System, siehe P2P-Seminar, Thema 2) und ICQ (Instant-Messaging, siehe Kap. 3) sind hierfür bekannte Beispiele.

### 2.4.3 Super-Peer Netzwerke

#### Eigenschaften der Super-Peer Netzwerke

Die so genannten Super-Peer Netze sind eine Mischform aus hybriden und reinen P2P-Netzwerken. In diesen Netzen werden bestimmte, sehr leistungsfähige Peers als zentraler Server für ein Cluster von Peers eingesetzt. Diese Peers werden Super-Peers genannt und verwalten und kontrollieren die Ressourcen der Peers in ihrem Cluster in einem Index und operieren damit als zentraler Server für ihr Cluster. Die Peers in einem Cluster können ihre Anfragen wie bei den hybriden Netzen an ihren Super-Peer stellen. Der Super-Peer durchsucht nun das gesamte Netz, indem er wie bei der Suche im reinen P2P-Netz die Anfrage weitersendet, jedoch nicht an alle Peers oder die Peers in seinem Cluster, sondern nur an diejenigen Super-Peers, mit denen er verbunden ist. Wird die Anfrage beantwortet, leitet der Super-Peer das Ergebnis an den anfragenden Peer in seinem Cluster weiter. Damit die Cluster nicht nur von einem Super-Peer abhängig sind, werden oft mehrere Super-Peers pro Cluster eingesetzt.

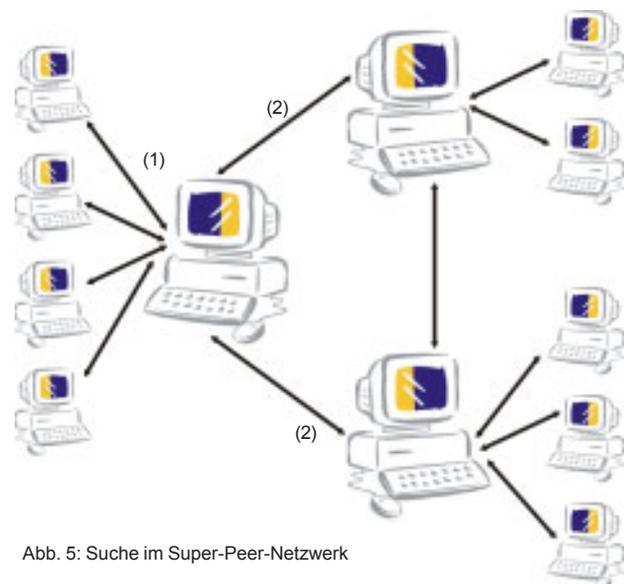


Abb. 5: Suche im Super-Peer-Netzwerk

(1) Suchanfrage an den Super-Peer (2) Super-Peers schicken die Anfrage an andere Super-Peers

#### Vorteile und Nachteile der Super-Peer Netze

Super-Peer Netzwerke vereinigen die effiziente Suche im hybriden Netzwerk mit der dezentralisierten Suche im reinen Netzwerk, durch die die Autonomie der Peers, die Robustheit des Netzes und auch die Lastverteilung im Netz besser unterstützt werden. Damit vereinigen sie die Vorteile beider Systeme. In einem reinen oder hybriden Netz haben außerdem alle Peers die gleichen Fähigkeiten, Möglichkeiten und Verantwortlichkeiten bei der Nutzung von Ressourcen, deshalb kann schnell die Performance des Netzes sinken, wenn die Anzahl der nicht so leistungsfähigen Peers steigt. Bei den Super-Peer Netzen werden nur besonders leistungsfähige Peers als Super-Peers ausgewählt, so dass die Leistungsfähigkeit des Netzes gesteigert wird.

Ein Problem besteht darin Peers zu finden, die sich dazu bereit erklären Super-Peers zu sein und ihre Bandbreite und Rechenleistung zur Verfügung stellen. Außerdem ist die Aktualisierung der Indizes aufwendiger, wenn mehrere Super-Peers pro Cluster eingesetzt werden.

Ein Beispiel für ein Super-Peer Netzwerk ist KaZaA.

### 3. Anwendungsbereiche im P2P

P2P steht noch relativ am Anfang der Entwicklung und es lässt sich heute kaum sagen, welchen Umfang, Einfluss und Möglichkeiten P2P-Anwendungen in der Zukunft haben werden. Doch schon jetzt gibt es eine Vielzahl von Anwendungsbereichen, die von P2P erobert werden und wo zum Teil herkömmliche Architekturen wie Client/Server Systeme schon ersetzt werden oder es entstehen neue Anwendungsbereiche. Die meisten Anwendungen kann man in eine oder mehrere der drei Gruppen

#### **Distributed Computing, Collaboration und Content Sharing (File-Sharing)**

einordnen.

Im Weiteren soll nur auf die Bereiche Distributed Computing und Collaboration näher eingegangen werden. Das Content-Sharing wird im Thema 2 des P2P-Seminars umfassend erläutert.

Kurz gesagt werden unter dem Begriff Content-Sharing bzw. File-Sharing viele verschiedene Bereiche zusammengefasst. Dazu gehören das Übertragen von Nachrichten, das Austauschen von Dateien, die verteilte Speicherung von Daten und damit das Teilen von Speicherressourcen, Caching, Knowledge Management, Content Management und damit in Verbindung stehende Dienste. Die Daten können dabei von ganz unterschiedlicher Art sein, Musik-, Video-, Bild-, Textdateien, Software und beliebige weitere Formate sind möglich, solange sie digitalisierbar sind. [BA02] Die drei größten technischen Herausforderungen beim Content-Sharing sind der Bandbreitenverbrauch, die Sicherheit, die Suche und verschiedene juristische Bedenken bez. der Daten.

#### 3.1 Distributed Computing

Nach dem Beratungsunternehmen Omni Consulting werden im Durchschnitt in Unternehmen nur 47% der Rechenleistung wirklich benutzt, wobei nicht nur Clients sondern auch Server und Workstations eingerechnet werden. [PF01] Bei privat benutzten PCs wird außerdem geschätzt, dass 80%-90% der Leistungsfähigkeit ungenutzt bleibt. [MI01] Auf der anderen Seite werden heute in vielen Unternehmen und in der Wissenschaft besonders leistungsfähige Supercomputer eingesetzt, um sehr komplexe und lange andauernde Berechnungen und Simulationen durchzuführen oder große Mengen an Daten zu speichern. Diese Supercomputer sind jedoch sehr teuer und schwer unterhaltbar und wartbar. Deshalb können sich viele finanzschwache Unternehmen diese Computer nicht leisten.

Mit dem Distributed Computing (DC - "Verteiltes Rechnen") wurde eine Lösung gefunden, die Nachfrage nach Ressourcen durch die freien Ressourcen, wie CPU-Zeit und Speicherkapazitäten, zu befriedigen. Die Nutzung von verteiltem Speicherplatz wird auch als Distributed Storage bezeichnet.

##### 3.1.1 Die Funktionsweise des Distributed Computing

DC-Anwendungen benutzen die Rechenleistung der Peers in einem internen (z.B. ein LAN in einem Unternehmen) oder externen (z.B. das Internet) Netzwerk um Aufgaben auszuführen, die normalerweise z.B. nur ein Supercomputer ausführen kann. Um dies zu bewerkstelligen, wird die CPU-Kapazität der Peers im Netz zum Berechnen von Teilaufgaben genutzt.

Ein zentraler Server zerteilt dazu eine komplexe Aufgabe in viele kleine unabhängige Teilaufgaben. Diese werden an die teilnehmenden Peers geschickt. Auf dem jeweiligen Peer ist eine Software

installiert, die die Teilaufgabe lokal ausführt und dann das Ergebnis zurück an den Server schickt, der dann wieder eine neue Teilaufgabe übertragen kann. Jeder Peer kann dabei bestimmen, ob seine Ressourcen nur verwendet werden sollen, wenn sein lokales System selbst gerade keine, bzw. kaum CPU-Leistung benötigt (z.B. wenn der Bildschirmschoner aktiviert wird, erkennt die DC-Anwendung, dass gerade kein anderer Prozess ausgeführt wird), also nicht alle Ressourcen ausgenutzt werden. Auf diese Weise bleiben die Peers selbst vollkommen autonom und einsatzfähig für ihre eigenen Aufgaben.

Um DC benutzen zu können, ist es unbedingt erforderlich, dass die komplexe Aufgabe wirklich in unabhängige Teile zerlegt werden kann, die keine Kommunikation zwischen den Peers erfordert. Deshalb ist die Nutzung von DC für Aufgaben, die normalerweise auf Rechnern mit z.B. mehreren parallelen Prozessoren ausgeführt werden, um z.B. Zwischenergebnisse immer weiter zu reichen, nicht möglich. Dafür sind Simulationen, bei denen immer wieder die gleiche Berechnung nur mit unterschiedlichen Parametern durchgeführt werden muss, besonders geeignet.

Viele Unternehmen und Institute konnten dieses Konzept schon erfolgreich anwenden und treiben die Entwicklung voran. Universitäten und Firmen benutzen die Leistungsfähigkeit der Arbeitsplatzcomputer ihrer Mitarbeiter um während der Nacht komplexe Berechnungen durchzuführen. DC-Anwendungen sind einsetzbar in Energieunternehmen, die vor Ölbohrungen geografische Analysen benötigen, Hersteller, die komplexe Strukturanalysen verrichten, Ingenieurunternehmen, die Belastungsproben ihrer Großprojekte durchführen, Unterhaltungs- und Grafikerunternehmen, die komplexe Bilder digitalisieren, Biotechnologiefirmen, die verschiedene Viren modellieren, Pharmaunternehmen, die neue Medikamente und Impfstoffe entwickeln und allen anderen Unternehmen, Hochschulen oder unabhängigen Projekte, die hochkomplexe Berechnungen durchführen müssen. [MI01]

In letzter Zeit haben verschiedene Projekte versucht funktionierende Geschäftsmodelle im Bereich des externen DC zu entwickeln, das heißt, dass nicht z.B. firmeninterne Rechner für das DC benutzt werden, sondern, dass man versucht Peers in externen Netzen anzuwerben. Peers werden hierbei je nach Umfang der zur Verfügung gestellten CPU-Zeit mit Geld oder Sachgütern bezahlt. In einem anderen Modell können Rechner ihre Leistung wie bei einer Auktion anbieten. Der Computer mit dem billigsten Angebot wird benutzt. Doch keiner dieser Ansätze scheint bisher richtig und vor allem wirtschaftlich zu funktionieren, so dass beiden beteiligten Seiten ein zufrieden stellender Nutzen entsteht.

Einige vertreten die Meinung, dass das DC nicht zu den P2P-Systemen gehört, da ja ein zentraler Server unbedingt nötig ist um die Teilberechnungen zu verteilen, zu verschicken und wieder zusammzusetzen und da die Peers nicht untereinander kommunizieren und auch selbst nicht als Server agieren. Trotzdem werden die Grundsätze des P2P erfüllt, dass Ressourcen der einzelnen Peers genutzt werden und diese trotzdem völlig autonom bleiben. Deshalb wird das DC im allgemeinen doch zu den P2P-Systemen gezählt.

Im folgenden Abschnitt soll eine der bekanntesten Distributed Computing Anwendungen vorgestellt werden.

### **3.1.2 SETI@home (SETI - Search for ExtraTerrestrial Intelligence)**

SETI@Home ist eines der bekanntesten Beispiele des Distributed Computing, bei dem Nachrichten von außerirdischen Lebensformen in empfangenen Radiowellen gesucht werden. Mehr als 4,5 Millionen Peers beteiligen sich derzeit, die innerhalb von 24 Stunden ca. 1000 Jahre CPU Zeit bei ca. 40-50 Teraflops (1 Teraflop entspricht 1000 Milliarden Fließkomma-Rechenoperationen pro Sekunde) zur

Verfügung stellen, indem sie ein kleines Programm als Bildschirmschoner ablaufen lassen.

Das Teleskop Arecibo in Puerto Rico sucht den Himmel in einem bestimmten Bereich nach Signalen ab, so daß pro Tag ungefähr 35 GByte Daten entstehen. Die aufgenommenen Daten müssen nun auf Unregelmäßigkeiten untersucht werden, dazu zerteilt man diese in sogenannte

work-units, die jeweils nur 340 KByte groß sind und den Peers zugeschickt werden. Die Bearbeitung der Daten wird offline durchgeführt. Nur um die fertig bearbeitete work-unit zurückzuschicken und eine neue zu empfangen, wird eine Verbindung aufgebaut. Work-units werden von mehreren Peers berechnet um sicherzustellen, dass genaue Ergebnisse erzielt werden und nichts verloren geht. [SE03]

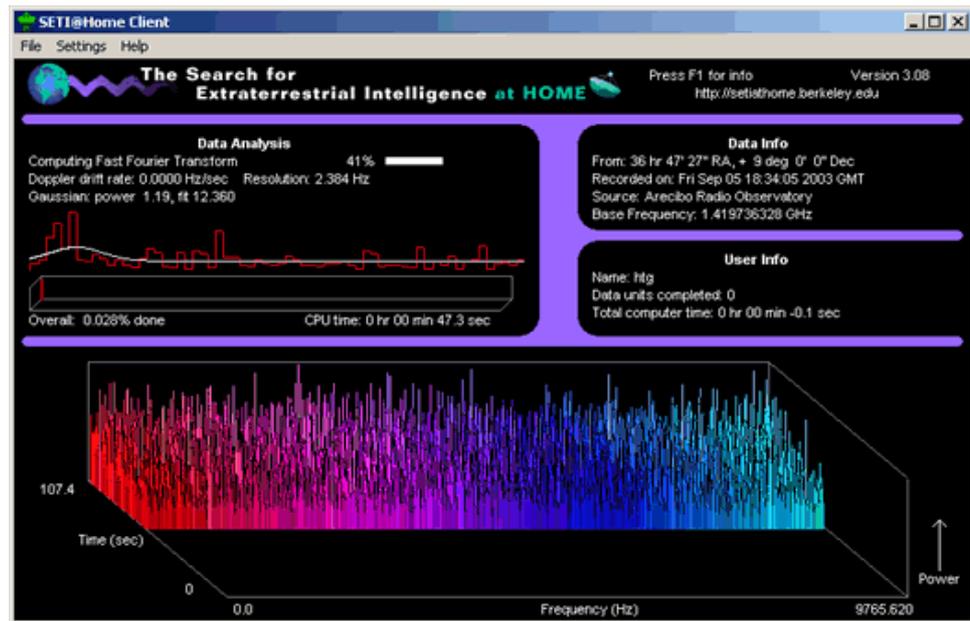


Abb. 6: SETI@home Anwendungsoberfläche

### 3.2 Collaboration

Menschen haben ein Bedürfnis nach Kommunikation mit ihren Freunden und Partnern. Über das Internet wurden in den letzten Jahren weitreichende Freundeskreise aufgebaut und Bekanntschaften geschlossen, mit denen man vor allem über E-mails oder Foren in Kontakt blieb. Mit der Zeit entstand jedoch der Wunsch schneller und direkter zu kommunizieren. Auch räumlich verteilte Teams in Unternehmen, die gemeinsam an Projekten arbeiten sollten, brauchten eine Möglichkeit schnell miteinander zu kommunizieren. Dafür reichte ein langsamer und umständlicher E-mail Dienst nicht mehr aus. Man brauchte Ad-hoc Lösungen. Die Collaboration-Anwendungen setzten diese Anforderungen um.

Collaboration kommt aus dem Englischen und bedeutet 'Zusammenarbeit'. Die P2P-Anwendungen, die diese Form des P2P unterstützen, gewährleisten, dass Peers in einem dezentralem Netzwerk in Echtzeit, also unmittelbar und ohne Verzögerung, kommunizieren und zusammenarbeiten können, ohne von einem zentralen Server oder internen Firmennetzwerk abhängig zu sein. Es steht also vor allem die direkte Kommunikation zwischen Menschen im Vordergrund, dies unterscheidet sie von den anderen Anwendungsbereichen. Trotzdem können Collaboration-Anwendungen die Übertragung von Inhalten und verteiltes Rechnen unterstützen.

Die Anwendungen basieren auf Ereignissen. Peers bilden eine Gruppe und beginnen eine Aufgabe gemeinsam zu bearbeiten. Dabei ist die Anzahl der Peers in der Gruppe unbedeutend. Wenn ein Peer eine Änderung durchführt (zum Beispiel möchte er eine neue Nachricht verschicken), wird ein Ereignis erstellt und an den Rest der Gruppe verschickt. In der Anwendungsschicht wird dann die Oberfläche der einzelnen Peers entsprechend aktualisiert.

### 3.2.1 Herausforderungen und Probleme

Hier soll nur auf zwei für die Collaboration typische Herausforderungen eingegangen werden, da die allgemeinen Anforderungen im Kapitel 5 näher betrachtet werden.

Eine besondere Herausforderung ist die **Fehlertoleranz**. Bei verteilten Anwendungen ist es besonders wichtig, dass alle Nachrichten sicher und verlässlich übertragen werden um zu gewährleisten, dass alle Peers die gleichen Informationen haben und die Daten konsistent bleiben, auch wenn sich Peers dynamisch im System bewegen, sich also auch ab- und anmelden. In den bisher existierenden Systemen, die nicht auf P2P basieren, entfallen diese Probleme durch die benutzte Client/Server Architektur. Im Bereich der P2P-Collaboration wird jedoch bisher noch nicht so genau darauf geachtet dieses Problem zu lösen. Nachrichten, die nicht übermittelt werden können, werden meist in einer Queue abgelegt und später übertragen, wenn dies wieder möglich ist.

Die Echtzeitübertragung von Daten und Informationen in einem P2P-Netz ist vielleicht die größte Herausforderung. Das Problem hierbei ist, dass dies nicht mit der Leistungsfähigkeit des P2P-Technologie zu tun hat, sondern mit dem darunter liegendem Netzwerk. In lokalen Netzwerken, z.B. innerhalb von Unternehmen, werden Anwendungen gut funktionieren, aber in weit verbreiteten Netzen wie dem Internet verlieren Anwendungen ihre Einsetzbarkeit durch hohe Wartezeiten und zu unterschiedliche Leistungsfähigkeiten der teilnehmenden Endgeräte. Spiele z.B. sind so aufgebaut, dass eine Aktion eines Spielers die virtuelle Umgebung verändert. Diese Veränderung wird an alle teilnehmenden Peers übertragen und erst wenn alle Computer die Daten empfangen haben, wird die Umgebung aktualisiert. Dies funktionierte in kleinen P2P-Netzen noch relativ gut aber im WAN werden Spiele unbrauchbar, da jede Aktualisierungsdauer von dem langsamsten Peer bestimmt wird.

### 3.2.2 Instant Messaging (IM)

Das Instant Messaging ist eine Unterklasse der Collaboration und der wohl bekannteste und weit verbreiteste P2P-Anwendungsbereich. Deshalb gibt es in diesem Bereich auch schon die ausgereiftesten Anwendungen.

Beim IM steht die Kommunikation zwischen den Peers im Vordergrund. Meldet sich ein Peer über eine Anwendung in einem Netz an, kann er mit Hilfe des Programms sehen, welche anderen User gerade auch im Netz angemeldet sind, wenn diese es zulassen (siehe Abb. 7). Nun kann der Peer durch das Versenden einer Nachricht, über einen einfachen Chat (Sonderform des IM, mehrere User können sich gleichzeitig unterhalten) oder einen Voice Chat (VoIP - telefonieren über das Netz) mit anderen angemeldeten Peers Kontakt aufnehmen. Einige Firmen nutzen das IM auch schon auf geschäftlicher Ebene zur Unterstützung der Kundenberatung auf ihren Webseiten. Durch die mangelnde Erfahrung in diesem Bereich und den erhöhten Kosten setzen aber nur wenige diese Lösung aktiv ein und der klassische E-mail Dienst wird weiterhin vorgezogen.

Bei den meisten Freeware-IM-Systemen kommt außerdem ein erhöhtes Sicherheitsproblem hinzu. Daten werden meist unverschlüsselt übertragen und die IM-Anwendungen besitzen Funktionen um herkömmliche Firewalls zu umgehen. Außerdem treten auch hier die für P2P üblichen Probleme auf, die in Kapitel 4 näher erläutert werden.



Abb. 7: ICQ Kontaktliste

ICQ, AIM oder der MSN Messenger sind drei viel genutzte IM-Systeme, die sich derzeit im Umlauf befinden.

### 3.2.3 Groupware

Die Groupware ist eine Form der Collaboration. Sie dient dazu die Kommunikation, Koordination und Zusammenarbeit von mehreren Menschen, die ein gemeinsames Projekt bearbeiten, zu erleichtern. Dafür besitzt Groupware alle Fähigkeiten und Möglichkeiten des Instant-Messaging und zusätzlich stellt es Funktionen zur Verfügung wie Event Planning, Kalender, Diskussionsforen, Project Management, Video Conferencing etc. Außerdem enthält sie Funktionen um erkennen zu können, ob eine bestimmte Person online ist, was sie bearbeitet oder mit wem sie gerade kommuniziert. Der besondere Vorteil von Groupware ist, dass ein unvermitteltes Kommunizieren und Kooperieren möglich ist und so sehr schnell auf sich ändernde Problemstellungen oder Situationen reagiert werden kann. Vor allem Unternehmen, deren Mitarbeiter sehr mobil sind oder geographisch verteilt arbeiten, die sehr dynamische Datenbestände oder sich schnell entwickelnde Geschäftsprozesse haben oder komplexe Beziehungen zu Partnern und Lieferanten pflegen, können viele Vorteile durch Groupware gewinnen.

Groupware an sich gibt es schon seit längerer Zeit, es wurden jedoch bisher klassische Client/Server Architekturen dafür eingesetzt. Unter Verwendung dieses Systems können jedoch Gruppen nicht beliebig zusammengestellt oder erweitert werden und sind immer von einem zentralen System abhängig. Mitarbeiter verschiedener Unternehmen konnten unter diesen Umständen nicht gemeinsam in einem Projekt arbeiten, da die meisten Systeme nur innerhalb eines Unternehmens funktionieren und jeweils sehr unterschiedlich aufgebaut sind. Mit der P2P-Groupware ist es möglich geworden, dass sich z.B. über das Internet jeder mit einer Arbeitsgruppe verbinden kann, ganz unabhängig von seinem Standort oder seinem lokalen Netzwerk.

Groove ist eine der bekanntesten Vertreter der Groupware-Systeme und soll kurz vorgestellt werden.

### 3.2.4 Groove

Groove bietet alle Funktionen, die eine Groupware unterstützen sollte (Foren, Texteditor, Dateiablage, Browser, Kalender, Chat, etc.) und legt dabei besonderen Wert auf die Anonymität der einzelnen Gruppenmitglieder. Gruppen werden gebildet, indem ein Benutzer einen Raum anlegt und andere Teilnehmer einlädt diesem Raum beizutreten. Nicht eingeladenen Peers ist es nicht möglich den Raum zu betreten. Der gesamte Raum wird auf dem Computer des teilnehmenden Peers verschlüsselt abgelegt. Jede Kommunikation sowie Dokumente werden über ein Public-Key Verfahren verschlüsselt und nur auf den Peers gespeichert. Groove übernimmt das Synchronisieren der Räume und ihrer Inhalte, wenn z.B. Änderungen offline durchgeführt wurden. Die Anwendung basiert auf offenen und allgemein akzeptierten Standards wie HTTP, UDP, TCP/IP und XML zur Datenspeicherung. Nachteile der Groove-Anwendung sind der große Speicherbedarf und dass das Systems relativ langsam ist.

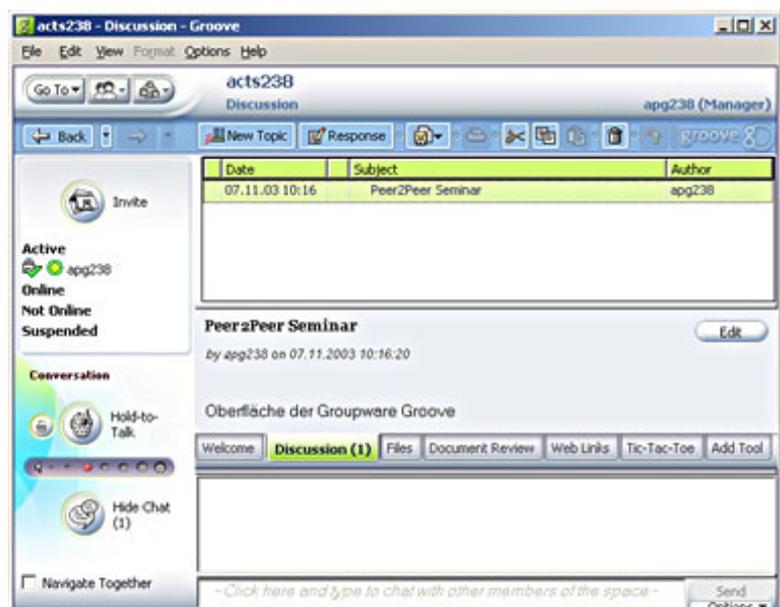


Abb. 8: Groove, Anwendungsoberfläche

## 4. Herausforderungen und Anforderungen an P2P-Systeme

P2P-Systeme haben durch die grundlegenden Konzepte des P2P von vornherein mit einigen Problemen und Herausforderungen zu kämpfen. Davon sollen im Folgenden die wichtigsten aufgeführt werden.

### 4.1 Technische Herausforderungen

#### 4.1.1 Heterogenität und Erweiterbarkeit

##### **Plattformunabhängigkeit / Heterogenität**

Endgeräten soll es möglich sein, sich jederzeit an einem P2P-Netz anzumelden und P2P-Anwendungen zu benutzen, ganz egal auf welcher Hardware sie aufsetzen, welche Software sie benutzen, auf welchem Betriebssystem sie laufen oder in welchem Netzwerk sie sich befinden. Auch die Art des Endgerätes soll ohne Bedeutung sein. Drahtlose Handhelds, Laptops, PCs, ganze Server Farms oder sogar Supercomputer sollen sich vollkommen gleichberechtigt anmelden können. Um diese Heterogenität zu unterstützen, muss eine P2P-Infrastruktur eine Vielzahl von Middleware Diensten für die unterschiedlichsten Endgeräte anbieten und modular und erweiterbar aufgebaut sein, damit sie auch bei neuen Entwicklungen schnell angepaßt werden kann. Eine P2P-Anwendung, die von vornherein nicht alle Plattformen unterstützt, wird sich kaum durchsetzen können.

##### **Interoperabilität**

Interoperabilität bedeutet, dass Peers unabhängig von der benutzten P2P-Anwendung und Infrastruktur und unabhängig von einem bestimmten P2P-Netzwerk miteinander kommunizieren können. Dies wird in den heutigen Systemen noch nicht durchgesetzt. D.h. es ist z.B. nicht möglich über ICQ mit jemanden Kontakt aufzunehmen, der einen MSN-Messenger benutzt. Auch für jedes File-Sharing System gibt es eigene Datenbanken, die die Informationen verwalten. Fast jede P2P-Anwendung besitzt also eigene Standards, Schnittstellen und Protokolle, so dass die Interoperabilität nicht gegeben ist. Nur einige Kommunikationsprotokolle und Austauschformate werden gemeinsam genutzt (SOAP, XML). Man ist jedoch auf dem Weg eine Standardisierung mit einheitlichen Schnittstellen und Protokollen zu entwickeln, die in allen P2P-Bereichen eingesetzt werden kann. Ein Beispiel für eine solche Infrastruktur ist JXTA von der Firma Sun.

##### **Skalierbarkeit**

Die Anzahl der teilnehmenden Nutzer in P2P-Systemen wächst ständig, deshalb muss die P2P-Architektur in hohem Grade skalierbar sein. D.h. sie muss unabhängig von der Anzahl der beteiligten Knoten funktionieren und es muss gesichert werden, dass einzelne Knoten nicht mit der steigenden Anzahl der Peers einer proportional steigenden Last unterliegen.

#### 4.1.2 Verwaltung von Ressourcen und Peers im P2P-Netz

##### **Gruppen Management** (siehe auch Thema 3 des P2P-Seminars)

Gruppen Management umfasst das Auffinden von anderen Peers in einem Netzwerk und das Routing zwischen den Peers. Dabei kann sich die Kommunikation nicht auf das DNS (Domain Name System) stützen, da normalerweise die Clients dynamisch zugeteilte IP-Adressen besitzen, die sich bei jeder Sitzung ändern können. Für die einzelnen Mechanismen muss man viele verschiedenen Faktoren beachten. Drahtlose, mobile Endgeräte können z.B. Peers nur finden, wenn sie innerhalb ihrer Reichweite sind. PCs benutzen für das Auffinden vor allem zentrale Verzeichnisse auf einem Server

mit fester Adresse, die immer aktualisiert werden sobald sich ein Peer im System an- oder abmeldet. Die Situation würde sich erheblich erleichtern, wenn alle Peers feste IP-Adressen hätten. Dies könnte theoretisch mit dem IPv6 realisiert werden, da dabei aber immer noch die Vergabe von dynamischen IP-Adressen erlaubt ist, wird dies in nächster Zeit nicht umsetzbar sein. Die Routing Algorithmen sollen den Weg einer Nachricht von einem Peer zum andern optimieren.

### **Performanz**

Da Peers in einem P2P-Netzwerk gleichberechtigt sind, kann jeder Daten senden und empfangen. Da aber bis vor kurzer Zeit asymmetrische Netze den Vorrang hatten, also Netze in denen relativ wenige Server Daten senden und viele Clients Daten empfangen wollten, wurden von den Anbietern von vorn herein für den Downstream größere Bandbreiten als für den Upstream bereitgestellt. Da nun aber viele Peers ihre Daten auch zur Verfügung stellen wollen, kann die Datenrate in diesen Netzen extrem absinken, denn die Geschwindigkeit, mit der man Dateien versenden kann, ist nie schneller wie die langsamste Anbindung der beteiligten Peers. Lösungen für dieses Problem bieten Caching Mechanismen um Daten nicht mehrmals hochladen zu müssen und die Möglichkeit Daten von mehreren Stellen simultan herunterzuladen.

### **Zuverlässigkeit / Persistenz**

Die dezentralisierte und dynamische Struktur der P2P-Netze macht es fast unmöglich die Zuverlässigkeit von Ressourcen zu gewährleisten. Jeder Peer kann z.B. Daten unbeabsichtigt oder beabsichtigt verschieben, löschen, ändern oder verfälschen und sich selbst an- und abmelden. Das beste Gegenmittel gegen die Unsicherheit bez. der Zuverlässigkeit ist die Redundanz. In File-Sharing Systemen können z.B. Dokumente auf vielen verschiedenen Peers redundant gespeichert werden. Beim Instant Messaging können Nachrichten bei Fehlern wiederholt oder über mehrere Wege gelenkt werden und beim Distributed Computing können Aufgaben auf verschiedenen Maschinen durchgeführt und die Ergebnisse verglichen werden. Um einen zuverlässigen Dienst anbieten zu können, muss z.B. sichergestellt werden, dass auch wenn der Peer sich abmeldet, die Daten fertig übertragen werden können.

### **Faire Ressourcenbereitstellung**

In P2P-Netzen zeigt sich ein erhebliches Problem, das Free-Riding. D.h. sehr wenige Benutzer stellen Ressourcen zur Verfügung, während sehr viele diese nur benutzen. Dies führt natürlich dazu, dass die Vorteile des P2P in den Hintergrund treten und die Nachteile der Client/Server Architektur auch hier entstehen.

## **4.1.3 Vertrauen, Sicherheit und Autonomie**

(Vollständigere Erläuterungen können im Thema 6 des P2P-Seminars nachgelesen werden)

### **Lokale Autonomie**

Auch wenn die Peers Ressourcen freigeben und teilweise den Zugriff anderer Peers auf ihr System erlauben, muss doch ihr lokales System autonom bleiben und unabhängig von anderen völlig selbstständig weiter arbeiten können. Jeder Dienst, der von der P2P-Middleware angeboten wird, muss diese Autonomie der einzelnen Peers respektieren. Der Benutzer des lokalen Systems soll allein darüber entscheiden können, wann, in welchem Umfang, wie und welche seiner Ressourcen er anderen zugänglich macht.

### **Lokale Sicherheitsbestimmungen**

Unter Umständen gehören Peers zu verschiedenen Teilnetzen, die untereinander kommunizieren aber verschiedene Sicherheits- und Administrationsmaßstäbe haben (z.B. Netzwerke verschiedener Unternehmen die Informationen nicht nach draußen geben wollen). Diese lokalen Bestimmungen

dürfen nicht überschrieben werden.

### **Vertrauen und Reputation**

Ein Peer muss anderen ihm meist unbekanntem Peers den Zugriff auf sein System und seine Ressourcen gewähren. Dazu braucht er Vertrauen, denn er erbringt dabei eine Leistung ohne rechtliche Sicherheiten zu haben. Deshalb müssen Maßnahmen vorgenommen werden um das Vertrauen zu fördern oder bei wirklich nicht vertrauenswürdigen Peers zu zerstören. Dazu können Sicherheits- und Reputationsmechanismen eingesetzt werden. Die Reputation eines Peers gibt Auskunft, wie er sich in der Vergangenheit verhalten hat. Da Peers den Verlust einer guten Reputation fürchten, der dazu führen kann, dass sie im Netz isoliert werden, halten sie sich im Allgemeinen an die Regeln.

### **Sicherheit**

Die Sicherheit in Netzwerken und besonders in P2P-Netzen ist eines der größten Probleme und Herausforderungen. Heute werden z.B. oft Daten einfach unverschlüsselt und ungeschützt übertragen. Viren können sich ähnlich wie bei E-mail Systemen rasend schnell über das ganze Internet verbreiten und Peers können dazu benutzt werden um z.B. Passwörter zu ermitteln. Beim Distributed Computing müssen oft kleine Anwendungen heruntergeladen werden, von denen man nicht weiß, ob sie u. U. Schaden anrichten können oder ob man ihnen überhaupt trauen darf.

Damit sich P2P aber auch im betrieblichen und kommerziellen Bereich durchsetzen kann, braucht man Anwendungen, die umfangreiche Mechanismen zur Sicherung wie die Authentifizierung, Autorisierung und Datenintegrität zur Verfügung stellen.

Um diese Mechanismen anwenden zu können, muss jedoch ein großer Aufwand durch den einzelnen Peer betrieben werden oder er muss sich mit einem vertrauenswürdigen Dritten verbinden. Oft kann dies nur erreicht werden, indem die Aufgabe der Sicherung zentralisiert wird.

## **4.3 Verwaltung von Ressourcen mit Hilfe von Datenbanken**

In den vorangegangenen Abschnitten wurden viele technische Herausforderungen aufgeführt. Besonders im Bereich der effizienten Suche, Speicherung und Verwaltung von Daten sind noch viele Probleme ungelöst und große Verbesserungen nötig. Dabei gibt es schon lange Mechanismen, die z.B. im Bereich der verteilten Datenbanken ein dezentrales Datenmanagement unterstützen. Diese können auch bei P2P angewendet werden. [GH\*01], [DGY03]

Nachfolgend soll die bisherige Umsetzung des Datenmanagements in P2P-Systemen dargestellt werden und gezeigt werden, welche Verbesserungen notwendig sind.

### **Optimierung der Datenhaltung**

In den heutigen P2P-Systemen hängt die verteilte Speicherung von Daten fast ausschließlich von der Nachfrage ab. Faktoren wie Bandbreite, Lastverteilung oder die Reputation der Peers spielen kaum eine Rolle. Deshalb kann man in den meisten Systemen nur auf die am häufigsten abgerufenen Objekte schnell zugreifen. Die Praxis zeigt jedoch, dass eine sehr hohe Anzahl an Anfragen Objekte sucht, die von fast keinem anderen gesucht werden (Zipfsches Gesetz).

Die Skalierbarkeit, Leistungsfähigkeit und Performanz eines Netzes kann verbessert werden, indem die Objekte und Objektverzeichnisse an strategischen Punkten abgelegt werden. Das Data Placement Problem spielt hier eine große Rolle. Es besagt, dass Daten und Arbeit so verteilt werden müssen, dass die gesamte Anfragebearbeitung mit den geringsten Kosten unter Beachtung der Bandweiten und Ressourcenbeschränkungen durchgeführt werden kann. Für die Optimierung von Suchanfragen und der Speicherung von Daten gibt es verschiedene Ansätze. Der erste Ansatz ist die Optimierung global über dem gesamten System durchzuführen. Dabei kann man das gesamte Wissen über die Datenverteilung, Lastverteilung, Bandbreitenbeschränkungen etc. ausnutzen. Dies ist aber sehr komplex, teuer und langsam, obwohl es zu einer sehr guten Optimierung führt. Andererseits kann

man jede Anfrage einzeln optimieren, was aber zu weniger guten Ergebnissen führt. Deshalb ist ein Mittelweg am besten.

### **Verbreitung der Objekte und Objektverzeichnisse**

Um eine gute Anfrageoptimierung durchführen zu können, braucht man eine Übersicht, welche Objekte zur Zeit der Anfrage auf welchen Peers gespeichert sind. Die einfachste Möglichkeit wäre alle Informationen über Objekte und ihre Speicherplätze in einem zentralen Verzeichnis abzulegen, dies würde aber wieder zu den Nachteilen einer zentralen Architektur führen. Eine zweite Möglichkeit wäre, das Verzeichnis auf allen Peers abzulegen, dadurch könnte man in minimaler Zeit und ohne zusätzliche Kommunikationskosten bei einer Anfrage gleich auf das Verzeichnis zugreifen, aber dieses Verzeichnis würde zuviel Platz benötigen und es würde ein riesiger Overhead entstehen, wenn die Daten nun auf jedem Peer aktualisiert werden müssten. Um die Vorteile beider Möglichkeiten zu verbinden, könnte man das Verzeichnis in Hierarchien aufteilen und die Teile unterschiedlicher Stufen auf mehrere Peers verteilen. Bei einer Anfrage sucht ein Peer nun zuerst bei einem ihm bekannten Peer. Findet er die gewünschten Objekte dort nicht, wird bei einem Peer angefragt, der eine Hierarchieebene weiter oben liegt. Bei all diesen Ansätzen ist es jedoch immer wichtig, dass die Konsistenz der Daten erhalten bleibt, egal ob Daten aktualisiert werden, sich die Datenverteilung ändert oder mehr Peers hinzukommen.

Eine andere Herausforderung ist, dass sich die Peers beliebig an- und abmelden können. Für die Optimierung bedeutet dies, dass man Daten so ablegen will, dass sie trotzdem zu jeder Zeit verfügbar sind. Wenn man nun Kopien von Daten auf Peers ablegen würde, die fast ununterbrochen angemeldet sind, wäre der Grundgedanke des P2P verletzt und die Leistungsfähigkeit würde sinken. Deshalb legen Systeme nur die originalen Dateien auf solchen statischen Peers ab und die Kopien werden auf vielen dynamischen Peers verteilt. Auf diese Weise ist die Erreichbarkeit stets gewährleistet, aber die Last verteilt sich weiterhin.

### **Datengranularität**

In einem P2P-System können Daten auf verschiedenen Granularitätsstufen abgelegt und abfragbar sein. Das *atomic granularity level* besagt, dass ein Datenobjekt aus mehreren untrennbaren Objekten besteht, z.B. eine mp3-Datei. Auf dieser Stufe muss man das ganze Datenobjekt auf einem Peer ablegen, man kann es nicht zerteilen. Dies unterstützen heutige P2P-Systeme. Bei dem *hierarchical granularity level* können mehrere Objekte zu einem übergeordneten Objekt zusammengefasst werden, so entstehen Hierarchien, z.B. können mehrere mp3-Dateien zu einem Album zusammengestellt werden und diese Alben können wieder in Kollektionen eingeordnet werden. Jetzt kann man entweder nur eine einzige Datei auf einem Peer ablegen oder das ganze Album. Die letzte Stufe ist die *value based granularity*. Dort werden Datenobjekte aus vielen atomaren (oder hierarchischen) Werten zusammengefügt. Z.B. Tupel in einer Relation bestehen aus Werten.

### **Erweiterung der Suchmöglichkeiten**

Bislang unterstützen P2P-Systeme nur sehr einfache Suchanfragen ohne Semantik und mit hoher Granularität, z.B. nach Schlüsselwörtern in den Namen von Dateien. Dies funktioniert jedoch nur bei atomaren Objekten, die umfassend und möglichst eindeutig durch ihren Namen oder eine Kennzeichnung beschrieben werden können (z.B. bei mp3-Dateien ist es meist völlig ausreichend nach dem Titel eines Musikstücks zu suchen). Eine Suche im Inhalt von Dateien, wie z.B. beim Information Retrieval oder die Suche nach sich überlappenden Objekten, wie es in Datenbanken möglich ist, wird nicht unterstützt. Um jedoch P2P-Systeme in vielen Bereichen einsetzen zu können, braucht man erweiterte Suchmöglichkeiten, z.B. um eine Rangfolge der zurückgelieferten Ergebnisse zu erhalten. Dazu bräuchte man wiederum bestimmte Aggregatfunktionen bei der Suche um z.B. die Anzahl von Dokumenten mit bestimmten Schlüsselwörtern zu finden. Es ist also unbedingt eine Erweiterung der Anfragesprache, ähnlich der Sprachen im Datenbankbereich, notwendig.

Damit jedoch auch ungelernete Benutzer schnell in einem P2P-System suchen können, muss die

erweiterte Anfragesprache möglichst einfach, für alle P2P-Systeme einheitlich und schnell erlernbar sein.

### **Korrektheit von Objekten**

Die Korrektheit von Objekten bedeutet hier nicht, dass Daten bei der Übertragung korrekt und vollständig beim Empfänger ankommen sollen. Es ist vielmehr damit gemeint, dass Mechanismen benötigt werden, um die (semantische) Konsistenz der Daten aufrecht zu erhalten. Z.B. soll man feststellen können, ob ein Objekt das unveränderte Original ist oder nur eine Kopie, die vielleicht verändert wurde. Es muss auch die Möglichkeit geben erstellte Objekte zu aktualisieren, wenn sie veraltet sind. Dies ist eine große Herausforderung, da viele Objekte gewollt im ganzen Netz redundant verstreut sind um die Antwortzeit und Verfügbarkeit zu verbessern, und dadurch Aktualisierungen dieser Objekte sehr aufwendig sind. Deshalb muss man einen geeigneten Grad der Verteilung von Objekten finden und gute Mechanismen zur Aktualisierung. Z.B. könnte jedes Objekt mit einem Ablaufdatum versehen sein oder ein zentraler Server könnte den Peers mitteilen, dass bestimmte Objekte nicht mehr aktuell sind. In den bisherigen P2P-Systemen wird die Korrektheit von Objekten nicht überwacht bzw. überprüft.

## **4.3 Juristische Probleme und Herausforderungen**

In einem P2P-Netz kann jeder Peer ungehindert und sehr einfach Inhalte erstellen, ändern und anbieten. Besonders bei dem dezentralen Modell (siehe Kapitel 3) werden diese Inhalte weder gefiltert noch kontrolliert oder zensiert. Illegale Inhalte wie z.B. Kinderpornographie können im Netz verbreitet werden und die Strafverfolgung wird durch die gegebene Anonymität der Peers und die Dynamik des Netzes erheblich erschwert. Nutzer können Dateien oder Dokumente, die mit einem Urheberrecht geschützt sind, einfach und schnell über das gesamte Internet verbreiten. Fertigt jemand für den privaten Gebrauch eine Kopie einer durch Urheberrecht geschützten CD an, mag dies nicht strafbar sein, wenn sich aber Millionen von Benutzern in einem Netz daran beteiligen, müssen ganz andere Maßstäbe gesetzt werden. Daten können auch verfälscht werden und Peers können nicht mehr sicher sein, ob sie z.B. ein Originaldokument erhalten oder ob die von ihnen erstellten Daten ungeändert weitergegeben werden. Dies führt zu einer Reihe von Problemen und juristischen Streitpunkten, die schon seit der Wiederbelebung des P2P für Diskussionsstoff sorgen und für die zum Teil auch schon neue Gesetze geschaffen wurden oder alte angewendet werden können. Lösungen wie Watermarking oder das Digital Rights Management erwiesen sich jedoch schnell als eher mangelhaft. Auch als das File-Sharing System Napster verboten wurde, führte dies nicht zum erwünschten Erfolg, da Benutzer schnell andere gleichartige Anwendungen und Möglichkeiten im Netz fanden.

Für all diese Probleme müssen erst noch Lösungen gefunden werden, die auch durchführbar sind.

Natürlich müssen sich nicht alle P2P-Anwendungen mit allen diesen Herausforderungen beschäftigen. In einem Firmennetzwerk braucht man sich z.B. keine Gedanken um die Verbindung weit entfernter Peers oder der Skalierbarkeit des Netzes zu machen, da die Anzahl der teilnehmenden Peers relativ feststeht und die Verbindung durch ein schon bereitstehendes LAN gewährleistet ist. Jedoch sollte eine wirkliche P2P-Infrastruktur diesen Herausforderungen trotzdem weitestgehend Stand halten. Andererseits gibt es auch spezifische Anforderungen für die einzelnen Anwendungsbereiche. Beim DC muss man sich z.B. über einen geeigneten Scheduling Mechanismus Gedanken machen oder Aufgaben so konzipieren, dass sie einfach in mehrere unabhängige Teilaufgaben zerlegt werden können. Beim File-Sharing braucht man bestimmte Datenstrukturen bzw. Datenbanken um die Informationen, nach denen gesucht wird, effizient abzulegen.

## 5. Zukunft des P2P und Zusammenfassung

Pat Gelsinger, Technischer Direktor von Intel sagte:

„Peer-to-peer computing could be as important to the Internet's future as the Web browser was to its past. While the most visible impact of this model has been in consumer environments, peer-to-peer computing has the potential to play a major role in business computing as well.“ [SN02]

Tim Andrews, Leiter des Innovationszentrums in Boston sagte schon 2000 voraus:

„P2P wird den Arbeitsablauf von Unternehmen und die Zusammenarbeit von Firmen mit ihren Kunden oder Partnern grundlegend verändern. Bis heute versuchen Mitarbeiter in den meisten Unternehmen, Wissen und Informationen anzuhäufen. P2P hingegen unterstützt den Austausch und die Weitergabe von Wissen. Durch engere Beziehungen auf geschäftlicher und persönlicher Ebene entstehen völlig neue Formen der Zusammenarbeit. Aus P2P werden interessante Geschäftsmodelle entstehen.“ [GE00]

Wie in den obigen Aussagen schon ausgedrückt, ist P2P eine wichtige Technologie geworden, aber man ist, anders als vielleicht in den obigen Aussichten erwartet, noch immer am Anfang der Entwicklung. Man wird erst langsam erkennen, welche Möglichkeiten man mit P2P überhaupt hat und in welchem Ausmaß sich P2P durchsetzen wird. Die in Kapitel 4 aufgeführten technischen, juristischen und sozialen Probleme und Herausforderungen müssen noch zufriedenstellend gelöst werden, standardisierte Protokolle und Architekturen sind notwendig, da man heute noch mehrere unterschiedliche Programme von verschiedenen Herstellern verwenden muss um z.B. File-Sharing, Distributed Computing und Instant-Messaging gleichzeitig betreiben zu können. P2P muss sich vor allem bei den Benutzern nicht nur im privaten, sondern auch im unternehmerischen und wirtschaftlichen Bereich durchsetzen, denn nur dadurch kann eigentlich, wie auch bei anderen Technologien, die wirtschaftliche Lebensfähigkeit sichergestellt werden. Für den Einsatz im professionellen Bereich muss außerdem ein geeignetes Geschäftsmodell entwickelt werden.

Für das Fortbestehen und die Weiterentwicklung von P2P-Systemen im größeren Maße spricht, dass sich große und marktführende Unternehmen wie Microsoft und Sun mit .NET und JXTA auf P2P einlassen, die Entwicklung vorantreiben und Standards schaffen wollen. Außerdem gibt es viele weitere industrieweite Projekte, Konsortien, Working Groups und Initiativen um das P2P zu verbessern und zu fördern.

Trotz all dieser Bemühungen und der guten Zukunftsaussichten setzen im Moment nur wenige Unternehmen P2P-Systeme ein, obwohl sie bei privaten Benutzern schon weit verbreitet sind. Millionen von Privatmenschen tauschen regelmässig innerhalb von P2P-Netzen Dateien aus. Die Schwierigkeit liegt nicht darin, dass die Unternehmen nicht daran interessiert sind, sondern dass es vor allem schwierig ist ein System zu finden um Ressourcen zu verwalten und den Datenaustausch zu kontrollieren. Außerdem können in einem P2P-Netz kaum QoS-Garantien gewährleistet werden, so dass für Unternehmen z.B. Arbeitsprozesse schwer kalkulierbar sind, da zu jedem Zeitpunkt eine andere Antwort auf eine Anfrage zurückgeliefert wird und es dabei zu unterschiedlich langen Bearbeitungszeiten kommt. Deshalb wird hier vielleicht versucht werden auf eine Kombination von P2P und Client/Server Architekturen aufzubauen. Im Allgemeinen kann man sagen, dass es in den Unternehmen zu einem Umdenken und Umstrukturieren kommen muss, ehe man diese neuen Technologien einsetzen kann, da ein Wechsel vom zentralen System zum dezentralen System vorzunehmen ist. Dies wird eine gewisse Zeit dauern. Deshalb wird nicht davon ausgegangen, dass das P2P die Client/Server Architekturen bald ablösen wird. Die Technologien sind dafür noch nicht ausgereift genug und für manche Aufgaben sind nach wie vor Client/Server-Systeme besser einsetzbar.

Trotzdem wird davon ausgegangen, dass sich P2P eine sichere Stellung auf dem Markt erobern wird, auch wenn es länger dauert als angenommen. Was die Zukunft wirklich bringt, wird sich zeigen müssen.

## Literaturverzeichnis

- [BA02] Barkai, D.: Technologies for Sharing and Collaborating on the Net  
Technology Research Labs, Intel Corporation, 2002
- [BR02] Becchio G., Reale D.: Peer-to-Peer, Revolution in der Kommunikation?  
Seminararbeit an der Fachhochschule bei der Basel (FHBB), Department Wirtschaft, 2002
- [DGY03] Daswani, N., Garcia-Molina, H., Yang, B.: Open Problems in Data-Sharing Peer-to-Peer  
Systems, Stanford University, Stanford CA 94305, USA, 2003
- [FR02] Frascaria, K.: Peer-to-Peer: Die Erneuerung des verteilten Rechnens  
ZDNet, <http://www.zdnet.de/itmanager/tech/0,39023442,2107183,00.htm>, 2002
- [GE00] Geraldts, J.: Intel group studies peer-to-peer technology  
<http://www.vnunet.com/News/1109593>, Silicon Valley, 2000
- [GH\*01] Gribble, S., Halevy, A., Ives, Z., Rodrig, M., Suci, D.: What Can Databases Do for Peer-to-Peer?,  
University of Washington, Seattle, WA USA, WebDB Workshop 2001
- [GY03] Garcia-Molina, H., Yang, B.: Designing a Super-Peer Network  
Stanford University, IEEE International Conference on Data Engineering, 2003
- [HE01] Heinrich, K.: Peer-to-Peer Datenbanken  
Proseminar: Multimediastandards im Internet, 2001
- [KL\*02] Kalogeraki, V., Lukose, R., Milojevic, D., Nagaraja, K., Pruyne, J., Richard, B., Rollins, S.,  
Xu, Z.: Peer-to-Peer Computing, HP Laboratories Palo Alto, HPL-2002-57, 2002
- [KR02] Krauß, H.: Konzeption und Realisierung der Server-Komponente für ein P2P-File-Sharing-  
System, bei dem die User am Umsatz beteiligt sind, Technische Universität Ilmenau, 2002
- [LM03] Luther, M., Meier, C.: Anwendungsgebiete von Peer to Peer Netzen  
Proseminar Funk- und Peer to Peer Netze, TU München, 2003
- [NS02] Nussbaum, M., Stohler R.: Peer-to-Peer, Revolution in der Kommunikation?  
Seminararbeit an der Fachhochschule bei der Basel (FHBB), Department Wirtschaft, 2002
- [PF01] Pfander, M.: Peer-to-Peer-Netzwerke: Ein Reizwort mit Potential  
InfoWeek 28/2001, [http://www.infoweek.ch/archive/a\\_single.cfm?artikelx=6083](http://www.infoweek.ch/archive/a_single.cfm?artikelx=6083), 2001
- [SS02] Schär, S.: Anwendungsgebiete von Peer-to-Peer Systemen  
Semesterarbeit zur Vorlesung Betriebssysteme, 2002
- [MI01] Miller, M.: Discovering P2P, Alameda CA (USA): Sybex Inc., 2001
- [SE03] SETI@home Website, <http://seti.alien.de>