

Informationen für die Lehrperson

GnutellaSim R2
Benno Baumgartner
27.09.2004

Worum es geht

Die Studenten sollen sich mit der Technik die hinter dem Schlagwort Peer-To-Peer steht beschäftigen. Dazu steht ihnen in erster Linie das Programm GnutellaSim zur Verfügung. Das Programm simuliert das Verhalten einer simplen Implementation des Gnutella Protokolls. Die technische Beschreibung des Protokolls finden Sie im Text gnutella_protocol_0.4.pdf[1]¹.

Wir sind der Meinung, dass ein Fachmann bzw. eine Fachfrau in der Lage sein soll über dieses Thema zu Diskutieren. Wir denken, dass es für die Peer-To-Peer Technik vor allem in Verbindung mit Grid computing [3] noch viele Möglichkeiten gibt.

Erforderliches Vorwissen

Konzept des Internets

Die Studenten wissen, dass im Internet jeder Computer über eine eindeutige Adresse verfügt. Es ist zudem bekannt, dass jeder Computer mit jedem anderen Computer eine „direkte“ Verbindung erstellen kann sofern die Adresse bekannt ist.

Konzept von Protokollen

Dass Computer ein gemeinsames Protokoll benötigen um miteinander zu kommunizieren ist bekannt. Ein intuitives Verständnis der Funktionsweise eines Protokolls genügt. Sollen die Studenten jedoch den Text gnutella_protocol_0.4.pdf[1] verstehen, ist ein tieferes Verständnis der Materie erforderlich. Dieser Text ist aber optional.

Technisches Englisch

Falls der Text gnutella_protocol_0.4.pdf gelesen werden sollte müssen die Studenten technisches Englisch verstehen.

Konzept von Paketen

Die Studenten wissen, dass Computer die miteinander verbunden sind und miteinander über ein Protokoll kommunizieren dies tun, indem sie Datenpakete austauschen.

Vorwissen das nicht Vorhanden sein sollte

Die Funktionsweise von Peer-To-Peer Netzen im allgemeinen und von Gnutella im speziellen muss nicht bekannt sein.

¹ Beachten Sie, dass der Simulator den PUSH REQUEST nicht simuliert. Der PUSH REQUEST wird verwendet um NATs (network address translators) und Firewalls zu umgehen und ist nicht von konzeptioneller Bedeutung. Für ein Nachrüsten des Simulators beachten Sie den Text developer_guide im docu Verzeichnis des Sourcecodes[2].

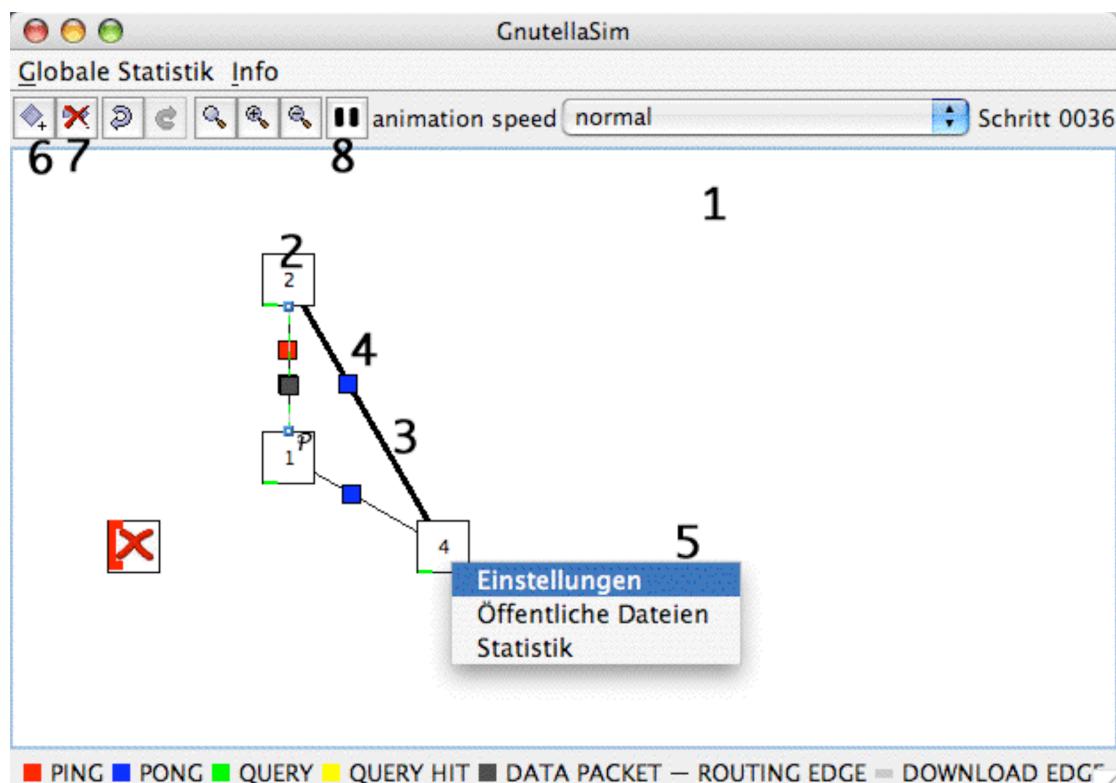
Ablauf

- Die Studenten lesen die Einleitung. (10')
- Die Studenten versuchen die Funktionsweise des Protokolls zu verstehen indem sie das Tutorial im Handbuch durchspielen und eventuell die technische Beschreibung des Protokolls lesen. (30')
- Die Studenten erstellen einen nicht technischen Vortrag über die Funktionsweise des Gnutella Protokolls. Der Vortrag soll sich an Laien richten. (60')
- Die Studenten tragen den Vortrag vor. (20')
- Die Studenten beschäftigen sich näher mit einem Aspekt des Protokolls den sie als interessant erachten. Sie beobachten das Verhalten des Netzwerkes und machen eventuell Verbesserungsvorschläge. Sie schreiben einen kurzen Bericht über ihre Erkenntnisse. (4h')

Wenn nicht so viel Zeit zur Verfügung steht, kann der Vortrag weggelassen werden.

Übersicht über GnutellaSim

Detaillierte Informationen finden Sie im Handbuch. Hier stellen wir Ihnen eine kurze Übersicht über das Programm zur Verfügung.



Figur1: Der Hauptbildschirm von GnutellaSim.

1. Zeichenfläche für den Graphen des Netzwerkes.
2. Ein Knoten im Netzwerk. Jeder Knoten verfügt über einen Buffer (grüner Balken). Läuft der Buffer über „stirbt“ der Knoten (dargestellt durch ein

- rotes Kreuz). Ein Knoten kann „aktiv“ oder „passiv“ sein. Ein Passiver Knoten (dargestellt durch ein P) erzeugt keine Nachrichten, leitet aber eintreffende Nachrichten weiter und beantwortet sie.
3. Eine routing Kante im Netzwerk. Über die routing Kanten werden Nachrichten (PING, PONG, QUERY, HIT) gesendet.
 4. Eine PONG Nachricht die vom Knoten 4 zum Knoten 1 gesendet wird.
 5. Kontextmenü eines Knotens das durch Rechtsklick auf einen Knoten aufgerufen werden kann. Über das Kontextmenü können Einstellungen für jeden einzelnen Knoten vorgenommen werden oder Statistiken über den Knoten abgefragt werden.
 6. Einfügen eines neuen Knoten. Die Einstellungen für den neuen Knoten entsprechen den Einstellungen, die Sie beim Programmstart im Wizard festgelegt haben.
 7. Nach dem Klick auf 7 können Sie neue Kanten erstellen, indem Sie die kleinen blauen Rechtecke (Ports) der beiden Knoten auswählen die Sie verbinden möchten.
 8. Starten bzw. stoppen der Simulation.

Nach dem Start der Simulation sieht man, wie die Pakete übers Netzwerk verteilt werden. Dabei entstehen neue Verbindungen und Dateien werden herunter geladen. Es wird empfohlen mit einem Netzwerk zu beginnen, das aus einer Liste von drei Knoten besteht, in der nur der erste Knoten aktiv ist. In einem solchen Netzwerk kann die Funktionsweise des Ping/Pong Verfahren am besten beobachtet werden. Bei diesem Verfahren sendet der aktive Knoten ein Ping per Broadcast ins Netzwerk. Jeder Empfänger sendet ein Pong zurück. Das Pong beinhaltet die Adresse des Senders. Das Pong wird über den Weg, den das Ping genommen hat zurückgeschickt (gerouted). Query und Hits funktionieren genau gleich.

Zur Analyse des Protokolls können auch grössere Netzwerke verwendet werden. Es stehen Analysewerkzeuge für jeden Knoten und für das ganze Netzwerk zur Verfügung².

Der Simulator eignet sich nicht für grosse Netze, da die Visualisierung viel Rechenleistung benötigt. Außerdem erfordern die Routingtabellen und die statistischen Informationen einen hohen Speicherbedarf. Es gibt Programme die grosse Netzwerke simulieren können, indem sie nicht in Echtzeit rechnen und komplett auf eine Visualisierung verzichten [4].

Mögliche Entdeckungen der Studenten

Die Liste ist nicht vollständig.

- Ein Knoten kann nur feststellen, ob jemand eine Datei herunterlädt, wenn die Datei von seinem Knoten heruntergeladen wird.
- Ein Knoten kann auf jede Suchanfrage antworten, egal ob er die Datei besitzt oder nicht. Der Knoten kann dadurch vortäuschen jede beliebige Datei zu besitzen. Dies wird dann besonders gefährlich, wenn der Knoten vorgibt ein Virus sei ein legitimes Programm.

² Das Hinzufügen neuer statistischer Informationen ist möglich. Beachten Sie hierzu bitte den Text developer_guide im docu Verzeichnis des Sourcecodes[2].

- Die wahrscheinlich einzige Möglichkeit das gesamte Netzwerk zu zerstören, ist jeden Knoten umzuprogrammieren, was praktisch unmöglich ist [5].
- Wenn die TTL für eine Suchanfrage sehr hoch gesetzt wird, erhöht sich die Chance etwas zu finden. Wenn das jedoch zu viele Knoten tun bricht das Netzwerk unter der Last der Queries zusammen, da diese per Broadcast verteilt werden.
- Dadurch, dass alle Anfragen per Broadcast verteilt werden ist das Netzwerk sehr ineffizient. Auch die Anforderungen an jeden Knoten sind hoch, da jeder Knoten eine Routingtabelle halten muss.
- Informationen über die Topologie des Netzwerkes zu sammeln ist aufwändig aber möglich.
- Jeder Knoten ist nur mit einer bestimmten Anzahl anderen Knoten gleichzeitig verbunden. Je grösser das Netzwerk wird desto schlechter wird das Verhältniss zwischen vorhandenen Knoten und verbundenen Knoten. Da Knoten ständig hinzukommen bzw. verschwinden kann es passieren, dass das Netzwerk in mehrere Teile zerfällt.
- Ein bestimmtes Layout kann erzwungen werden. Es kann zum Beispiel eine Art Fat Tree Struktur erzeugt werden, wenn Knoten mit hoher Bandbreite nur mit Knoten mit ähnlich hoher Bandbreite Verbindungen eingehen. Dies funktioniert nur, wenn sich eine grosse Zahl von Knoten an dieses Prinzip halten. Da jeder Knoten das Interesse hat mit einem möglichst schnellen anderen Knoten zu verbinden, ist dies schwierig durchzusetzen.
- Ein grundsätzliches Problem ist, dass man darauf vertrauen muss, dass sich eine überwiegende Zahl von Knoten an die Spezifikation halten. Man hat keine direkte Möglichkeit dies festzustellen. Man kann Heuristiken entwickeln die es erlauben verdächtige Knoten zu erkennen, um die Verbindungen zu ihnen zu trennen.

Die Studenten könnten Optimierungen für die folgenden Parameter entdecken

- Jeder Knoten speichert Queries und die entsprechenden Hits über eine bestimmte Zeit. Empfängt ein Knoten ein Query mit demselben Inhalt leitet er das Query nicht weiter sondern antwortet sofort mit dem gespeicherten Hit.
- Jeder Knoten speichert Pongs die er erhält und baut eine Tabelle auf. Wenn der Knoten eine neue Verbindung braucht, sendet er kein Ping, sondern schaut in seiner Tabelle nach.
- Ein Knoten kann eine Datei nicht nur von einem Knoten herunter laden, sondern verschiedene Teile der Datei von mehreren Knoten gleichzeitig (Swarm Download).
- Knoten nur mit Knoten verbinden, die eine ähnliche Bandbreite besitzen.

Die Studenten könnten folgende Protokolle entwickeln

- Es gibt zwei Arten von Knoten: Normale Knoten und super Knoten (Supernodes). Supernodes verfügen über eine sehr hohe Bandbreite und bilden ein eigenes Peer-To-Peer Netzwerk untereinander. Jeder normale Knoten ist mit genau einem Supernode verbunden. Jeder Supernode kennt alle Dateien von den normalen Knoten die mit ihm verbunden sind. Eine Suchanfrage geht an den Supernode. Besitzt keiner der angeschlossenen Knoten die Datei fragt der Supernode die anderen Supernodes nach dieser Datei. Besitzt ein anderer Supernode die Datei, speichert der anfragende Supernode diese Information und leitet sie an den anfragenden Knoten weiter.

- Ein Knoten lädt nicht direkt von einem anderen Knoten eine Datei herunter sondern fragt einen beliebigen anderen Knoten an, ob dieser die Datei herunterladen kann. Dieser lädt dann die Datei vom Besitzer hinunter. Der anfragende Knoten lädt dann die Datei vom angefragten Knoten herunter. Dadurch kann der Knoten der die Datei rauflädt nicht feststellen wer die Datei hinunterlädt. Aber der Knoten der hinauflädt ist bekannt. Zusätzlich kann Verschlüsselung eingesetzt werden, dann weiss der mittlere Knoten nicht, was er transportiert.
- Ein Hit besitzt nicht die Information welcher Knoten die Datei besitzt. Stattdessen fragt man den Knoten von dem man den Hit gekriegt hat, ob er die Datei herunterladen kann. Dieser wiederum fragt den Knoten von dem er den Hit gekriegt hat und so weiter. Die ganze Datei wandert dann den Pfad entlang den auch die Hit Nachricht genommen hat. Dadurch sind sowohl Uploader wie auch Downloader unbekannt. Der Effizienzverlust ist allerdings enorm [6].

Prüfung, Bewertung von Berichten, Bewertung von Vorträgen

- Der Vortrag ist korrekt und leicht verständlich. Nicht korrekt ist z.B.: Wenn man eine Datei sucht fragt man einen Server wer die Datei besitzt. Gibt es jemanden lädt man die Datei von ihm herunter.
- Die Studenten können in maximal 10 Sätzen einem Experten erklären, wie das Protokoll funktioniert. Die Studenten wissen:
 - Dass das Protokoll auf Broadcasting basiert.
 - Dass ein Download nicht über das Netzwerk stattfindet sondern direkt zwischen zwei Knoten.
 - Dass der einzige Zweck des Netzwerkes darin besteht verteilt nach Dateien zu suchen.
- Die Studenten können mindestens ein Problem des Protokolls beschreiben und belegen. Das Problem muss Protokollspezifisch sein. Eine Aussage wie: „Wenn weltweit der Strom zusammenbricht, bricht auch das Netzwerk zusammen.“ gilt nicht. Das Problem ist durch Tabellen oder Graphiken dokumentiert und ist nachvollziehbar.
- Die Studenten können Protokollanpassungen und Protokollverbesserungen formulieren oder glaubwürdig darlegen, dass eine Lösung nicht möglich ist. (Lösungen die nur dann funktionieren, wenn jeder Knoten über eine 100 Mbit/s Anbindung verfügt sind nicht umsetzbar.)

Referenzen:

- [1] gnutellasim2.sourceforge.net
- [2] sourceforge.net/projects/gnutellasim2/
- [3] gridcafe.web.cern.ch/gridcafe/
- [4] peersim.sourceforge.net
- [5] http://www.heise.de/tp/deutsch/inhalt/te/11270/1.html
- [6] freenet.sourceforge.net