
Diplomanden- und Doktorandenseminar

Implementierung eines Gnutella-Clients für IPv6

1. Motivation
2. IPv6
3. Gnutella
4. Portierung

Frank Sowinski

17.12.2002



Motivation

Gute Gründe für IPv6

- ❑ Das Anwachsen des Internets führt zu einer deutlichen Erhöhung der Netzbelastung und einer Verknappung der freien IP-Nummern.
- ❑ Neue Anwendungen erfordern neuartige Dienste und Funktionalitäten
 - z.B. die Synchronisation von Video- und Audiodaten.
- ❑ Hohe Datenraten erfordern geeignete Paketformate zur effizienten Bearbeitung.
- ❑ Der Einsatz von immer mehr mobilen Geräten.

→ Programme sollten an die neue Socket-API angepaßt werden.

- ❑ Exemplarisch: Portierung von Gnutella



IPv6



IPv6-Header

40 Bytes, 8 Felder, feste Länge (RFC 2460)

Version	Traffic Class (8 Bit)	Flow Label (20 Bit)	
Payload Length		Next Header	Hop Limit
IP Source Address (128 Bit)			
IP Destination Address (128 Bit)			

— neu: Quality-of-Service-Unterstützung



IPv6 Erweiterungsheader

- Sechs Kopferweiterungen. Empfohlene Reihenfolge:
 1. Hop-by-Hop Options
 2. Destination Options (1) für Zwischenziele
 3. Routing Header
 4. Fragment Header (Ende-zu-Ende-Fragmentierung)
 5. Authentication Header für Authentifizierung/Integritätssicherung
 6. Encapsulating Security Payload für Verschlüsselung
 7. Destination Options (2) für endgültiges Ziel

IPv6-Kopf Next Header = Routing	Routing-Kopf Next Header = TCP	TCP-Kopf + Daten
------------------------------------	-----------------------------------	---------------------



Gnutella



Gnutella Protokoll

(1)

Allgemein:

- ❑ Dient dem Download von Dateien aus dem Netz (File Sharing).
- ❑ Arbeitet ohne zentralen Server.
- ❑ Ist sowohl SERVER als auch cliENT \Rightarrow Servent oder auch Servant.
- ❑ Einsatz von einer GUID/UUID zur Identifizierung/Routing.
- ❑ Peer-to-Peer (P2P)-Anwendung (Peer = Gleicher).
 - Das Programm kann mit dynamischen Verbindungen/temporären Netzadressen umgehen.
 - Die beteiligten Endsysteme am Rand des Netzes sind größtenteils autonom.
- ❑ Großes, fehlerredundantes Netz.

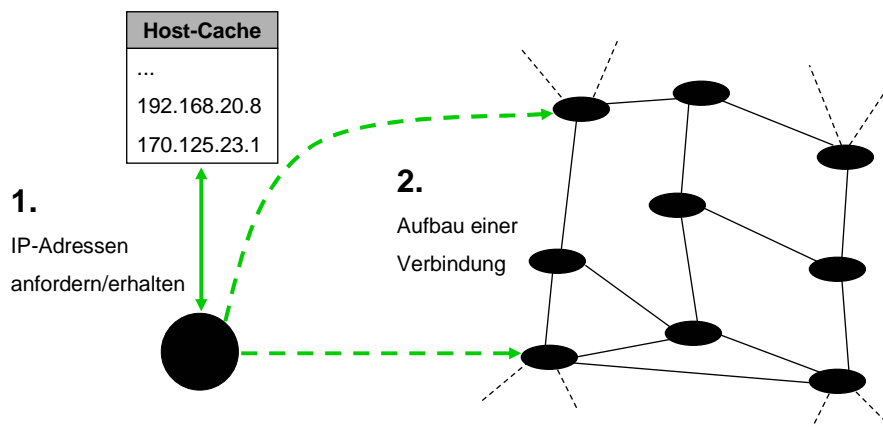


Gnutella Protokoll

(2)

1. Verbinden mit dem Gnutella-Netz (Bootstrapping)

- ❑ Verbindung mit vordefinierten und permanenten Host-Caches.
- ❑ Servent erhält IP-Adressen von Peers.
- ❑ Versuch eines Verbindungsaufbaus.

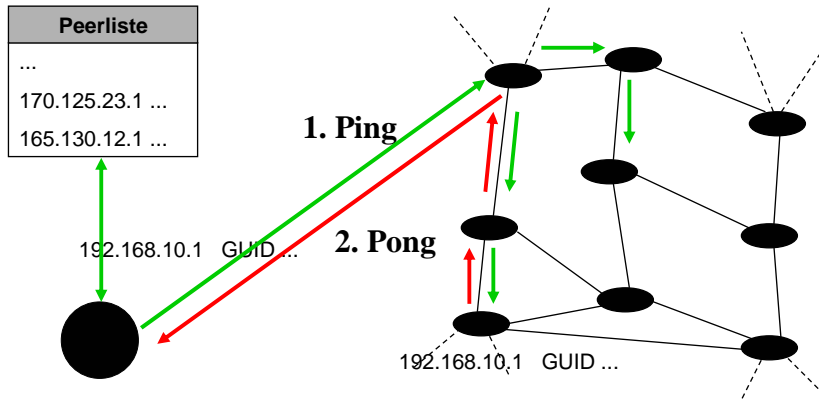


Gnutella Protokoll

(3)

2. Entdecken von neuen Gnutella-Peers

- ❑ Verbindungen sind nicht permanent \Rightarrow neue Peers werden benötigt.
- ❑ Peer broadcastet periodisch Ping-Nachrichten.
- ❑ Pong-Nachricht eines aktiven Peers \Rightarrow dynamische Liste wird erweitert.
- ❑ Ggf. Verbindungsaufbau.



Implementierung eines Gnutella-Clients für IPv6

www.ibr.cs.tu-bs.de

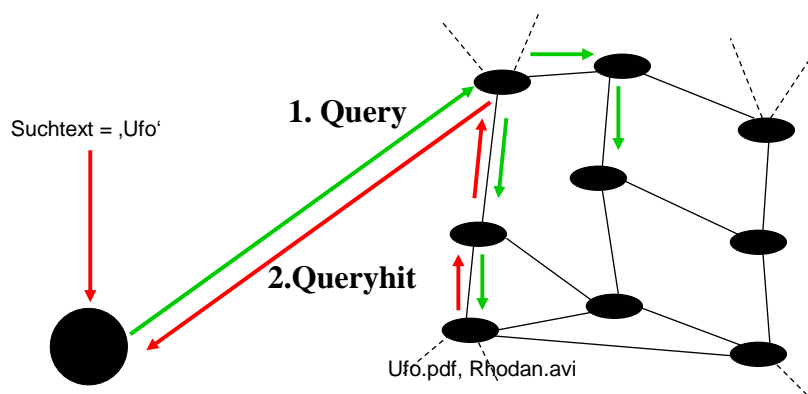


Gnutella Protokoll

(4)

3. Suche nach Dateien

- ❑ Peer broadcastet Query-Nachricht (meist TTL=7).
- ❑ Empfänger durchsucht seine lokale Dateien.
- ❑ QueryHit-Nachricht wird bei einem Treffer zurückgesendet.



Implementierung eines Gnutella-Clients für IPv6

www.ibr.cs.tu-bs.de

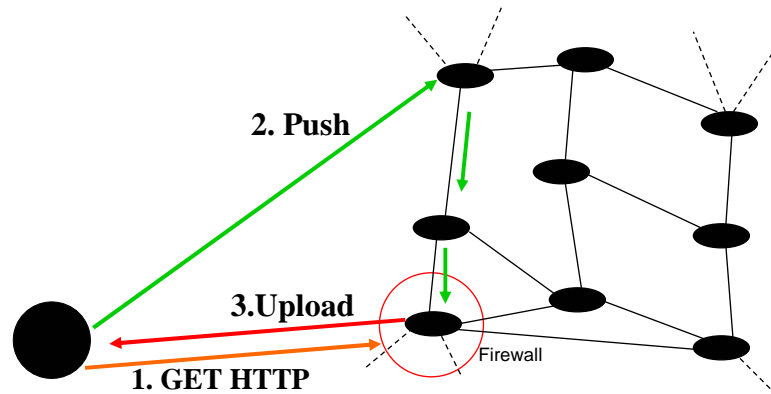


Gnutella Protokoll

(5)

4. Download von Dateien

- ❑ Direkt durch ,GET HTTP'-Anfrage.
- ❑ Indirekt durch Push-Nachricht an Anbieter hinter einer Firewall.
- ❑ Problem: beide Peers hinter einer Firewall.



Implementierung eines Gnutella-Clients für IPv6

www.ibr.cs.tu-bs.de



Skalierbarkeit

1. Gnutella-Generation

- ❑ Schmalbandige Peers konnten die Last nicht mehr bewältigen.
 - z.B. 56K-Modem bei 1000 Knoten mit Ping- und Query-Nachrichten überlastet.

2. Gnutella-Generation

- ❑ Anzahl der Verbindungen werden Bandbreite des Peers angepaßt.
- ❑ Verbindung zu überlasteten Peers wird abgebrochen.

3. Gnutella-Generation

- ❑ Einführung von Hierarchien \Rightarrow keine reine P2P-Anwendung mehr.
- ❑ Ultrapears übernehmen Netzlast für schmalbandige Peers.

Implementierung eines Gnutella-Clients für IPv6

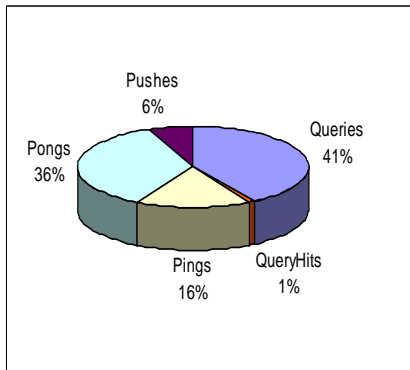
www.ibr.cs.tu-bs.de



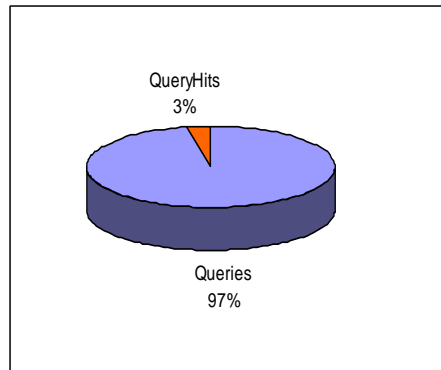
Analysen

(1)

Gnutellaverkehr (typisches Beispiel)



Lastverteilung



Erfolgreiche gegen erfolgreiche Queries

➔ Großer Overhead, geringe Ausbeute!



Analysen

(2)

Analysen haben weiter ergeben:

- ❑ Das Netz bestand im Mai 2001 aus etwa 50.000 Knoten.
- ❑ Ungefähr 40% der Knoten verlassen das Netz nach weniger als vier Stunden.
- ❑ Ungefähr 25% der Knoten bleiben für mehr als 24 Stunden im Netz.
- ❑ 95% aller Knotenpaare können Nachrichten über weniger als 7 Hops austauschen ⇒ fast alle Broadcast-Nachrichten erreichen fast alle Knoten.
- ❑ Die mittlere Pfadlänge wächst logarithmisch mit der Netzgröße.
- ❑ TCP-Broadcast ⇒ die benötigte Bandbreite pro Request wächst jedoch linear mit der Netzgröße.



Gefahren

Typische Gefahren im Gnutella-Netzwerk:

- ❑ Distributed Denial of Service (DDoS) durch QueryHit mit falscher IP-Adresse (Spoofing).
- ❑ Pong-Attacke mit falscher IP-Adresse (wegen periodischen Pings nur temporär).
- ❑ IP Harvesting.
- ❑ Übermittlung von Viren durch Push-Nachricht.
- ❑ GUID Tracing.
- ❑ Eingebaute Spyware in Gnutella-Clients.



Portierung



Für die Portierung kritische Nachrichten

Pong-Nachricht

Port	IP-Adresse	TTL	Hops
2 Byte	4 Byte	4 Byte	4 Byte

Kritische Nachrichten (haben eine 4 Byte lange IP-Adresse):

- Pong
- Push
- QueryHit

Neue Nachrichten brauchen 16 Bytes für IP-Adressen.

➔ Alle Servents müssen neues Format kennen!



Schlußfolgerungen für die Portierung

Positiv:

- Socket-API bietet Funktionen für protokollunabhängige Programme (siehe [2] + [3]).

Negativ:

- Gnutella-Protokoll ist auf IPv4-Adressen ausgerichtet.



Literatur

- [1] Christian Huitema: IPv6 die neue Generation, Addison-Wesley, 2000
- [2] W. R. Stevens: Advanced Name and Address Conversions,
<http://www.kohala.com/start/unpv12e.html>
- [3] W. R. Stevens u.a.: RFC 2553: Basic Socket Interface Extensions for IPv6,
<http://rfc.sunsite.dk/main.html>
- [4] Martina Zitterbart: Next Generation Internet: Kapitel 9: Peer-to-Peer (P2P),
www.tm.uka.de, 2001
- [5] Gnutella Protocoll Specification v0.4 u. v0.6, <http://rfc-gnutella.sourceforge.net/>

