Freenet, Gnutella, Napster

Moderne Methoden der dezentralen
Informationsspeicherung und -verbreitung

Dipl. Inform. Arno Wagner
Communication Systems Group, ETH Zürich

Vortragsübersicht

1. Zentralität/Dezentralität am Beispiel bekannter Protokolle
2. Warum Dezentralität erstrebenswert ist
3. Beispiel I: Napster
4. Beispiel II: Gnutella
5. Beispiel III: Freenet

Zentralität/Dezentralität am Beispiel

- IP-Protokoll:
  - Zentrale IP-Adressen-Vergabe
  - Teilweise hierarchische Routing-Informationen
  - Keine **dynamischen** zentral verwalteten Daten
- DNS:
  - Hierarchische Struktur, von Root-Servern abhängig
  - Durch lokales Caching begrenzte Überlebensfähigkeit bei Ausfall der Root-Server
- WWW:
  - Benutzt IP-Protokoll
  - Ist in der Praxis meist auf DNS angewiesen

Vorteile von Dezentralität

- **Zuverlässigkeit**
  - Keine Single-Point-Of-Failure's
- **Leistungsfähigkeit**
  - Leistung nicht durch zentrale Infrastruktur begrenzt
  - Bessere Skalierbarkeit
- **Überlebensfähigkeit**
  - Technische/juristische Attacken stören i.d.R. nur Teile des Systems
  - Redundanzen können Ausfälle kompensieren
- **Anonymität**
  - Teilnehmer benötigen keine (zentral) bekannte Identität

Nachteile von Dezentralität

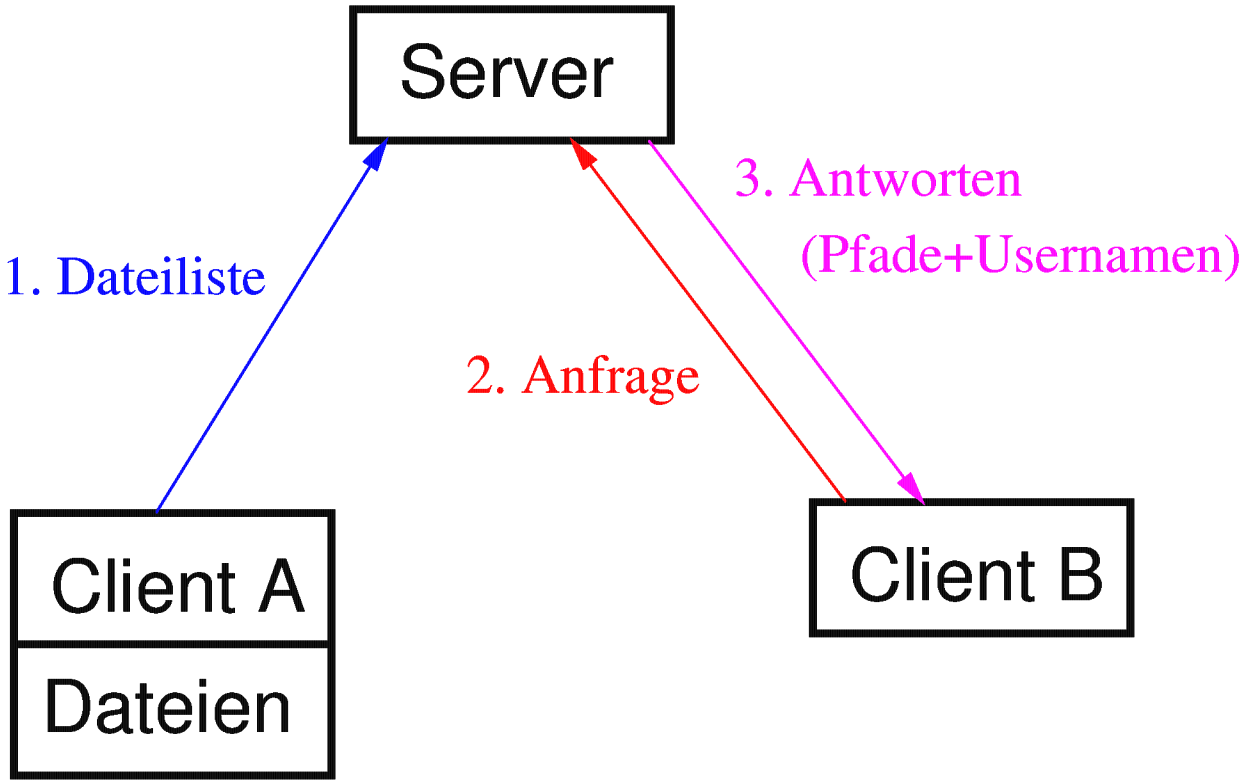
- **Unübersichtlichkeit**
  - Schwierigere Systemdiagnose bei Problemen
  - Statistiken (Teilnehmerlisten, Effizienz, Systemlast,...) schwerer ermittelbar
- **Unkontrolierbarkeit / Missbrauchsanfälligkeit**
  - Korrektur von Problemen schwierig
  - Unerwünschte Aktivitäten schwer erkennbar/verhinderbar
  - Keine verantwortlichen Personen vorhanden
- **Ineffizienz**
  - Summe des Verwaltungsaufwandes höher (aber verteilt)
  - Vielfach replizierte Daten

Beispiel I: Napster

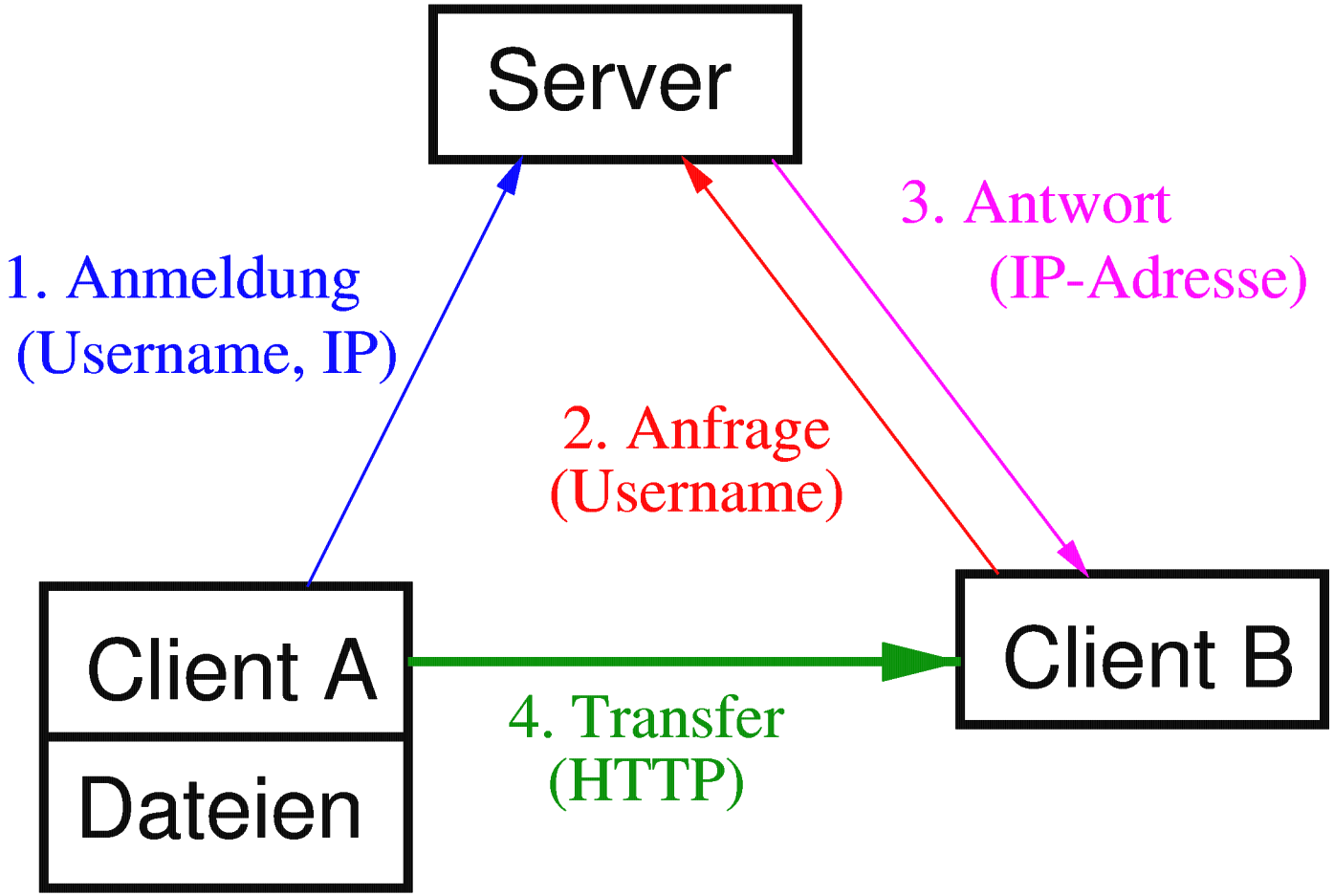
- Kommerzielles Produkt von Napster, Inc.
  Freie Alternativen verfügbar (http://opennap.sourceforge.net)
- Server Liste unter http://www.napigator.com/list.php
- **Ziele/Zweck:**
  - Sharing von mp3-Dateien mit Suche in zentraler Datenbank
  - Business Model unklar
    Neue Entwicklung: Allianz mit Bertelsman, Lizenzkosten
- Online seit ca. 16.11.1999
- Client-Server basiert
  - Client: Für diverse Plattformen (ursprünglich nur MS Windows)
  - Server: Orginal nicht verfügbar, freie clones für Unix

Ablauf: Suche

Suchmöglichkeit: Substring



Ablauf: Dateitransfer



Anonymität

- **Server**
  - Kennt Username und benutzte IP-Adressen
  - Kennt alle jeweils angebotenen Dateien
  - Kennt alle Dateitransfer*versuche*
- **Client**
  - Anbieter
    * Keine Informationen bzgl. Suchen
    * Bei Download: IP-Adresse des Abnehmers
  - Abnehmer
    * Bei Suche nur Username und Pfad des Anbieters sichtbar
    * Bei Download IP-Adresse des Anbieters bekannt

Anonymität: Mäßig gegenüber Teilnehmern, keine gegenüber Server

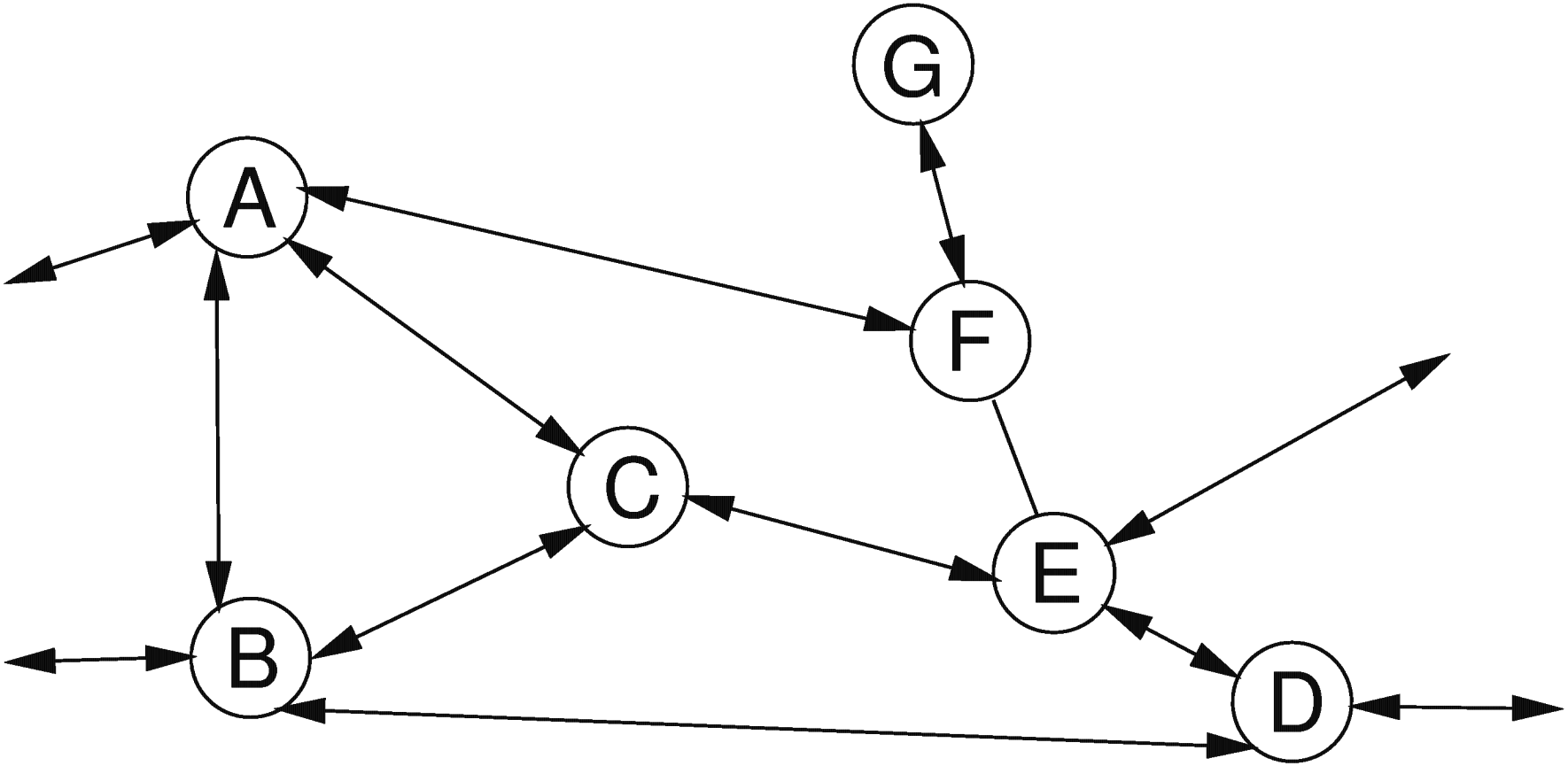
Fazit Napster

- Kommerzielle Fundierung, mittlerweile nichkommerzielle Alternativen
- Zentralistische Struktur, zentrale Kontrolle möglich (z.B. Zugang)
- Technische / juristische Attacken auf Server reltiv einfach
- Flaschenhals: Leistung der Server

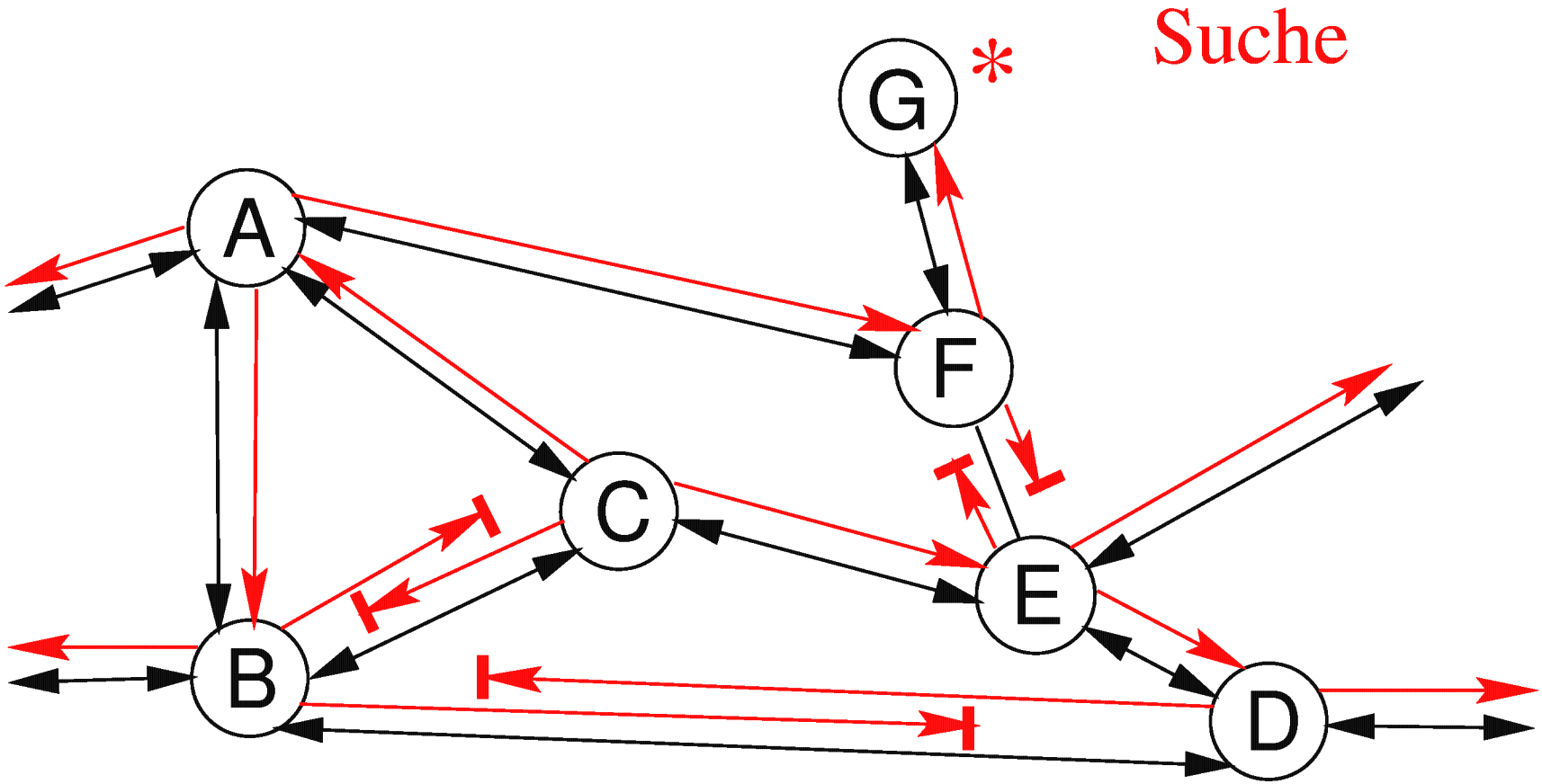
Beispiel II: Gnutella

- „Nachfolger“ von Napster, unter GPL
- Ziele/Zweck:
  - Sharing von beliebigen Dateien mit verteilter Suche
  - Beseitigung des Server-Flaschenhalses von Napster
  - Beseitigung der „Zensierbarkeint“ von Napster?
- Publiziert von Nullsoft: Source am 14.3.2000 einige Stunden im Web, dann von AOL/Time Warner unterdrückt
- Freie Weiterenwicklung unter GPL (http://gnutella.wego.com)
- Client = Server, Selbstorganisiertes Netz
- Diverse Platformen, initial nur MS Windows

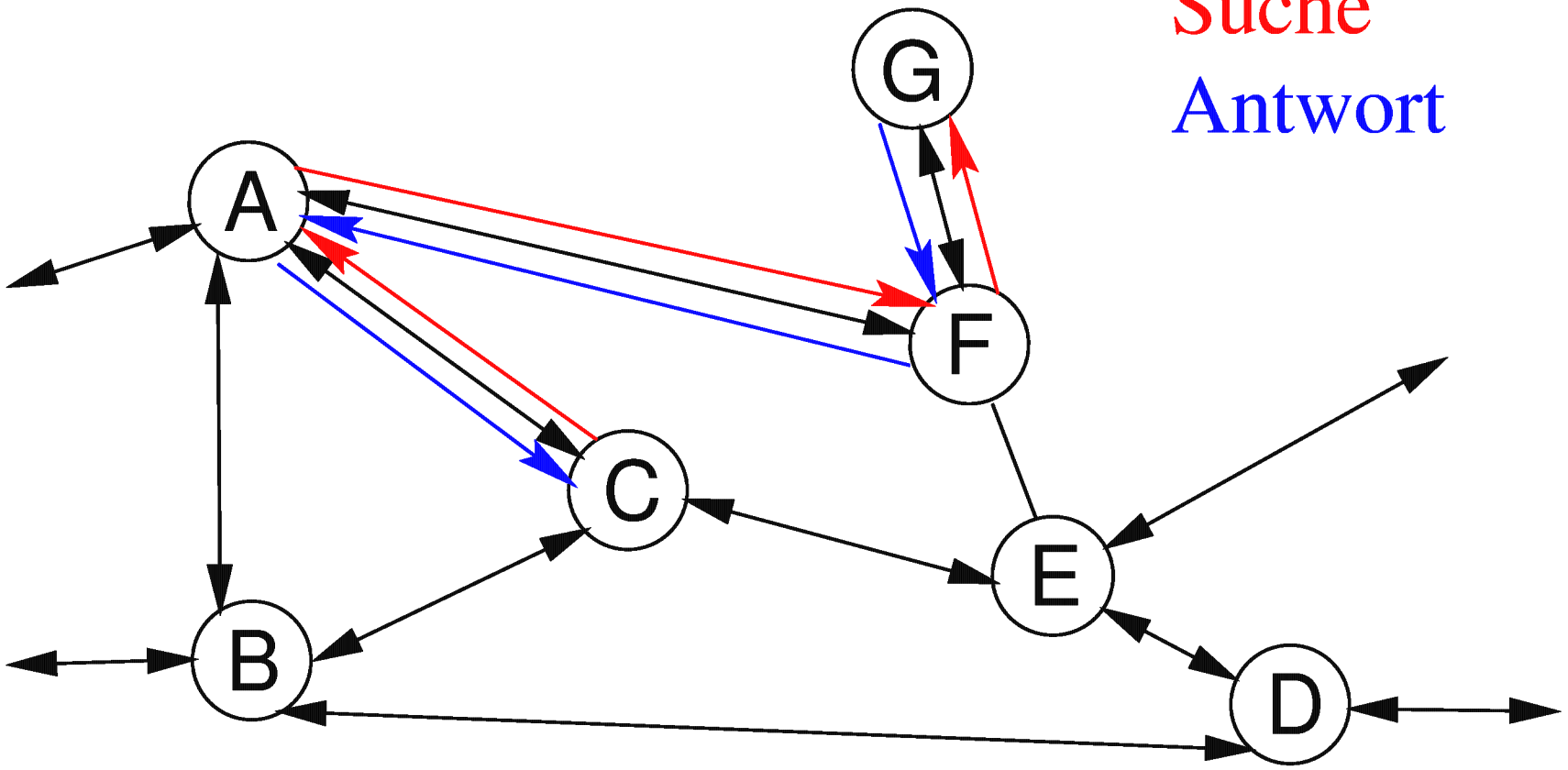
Beispieltopologie



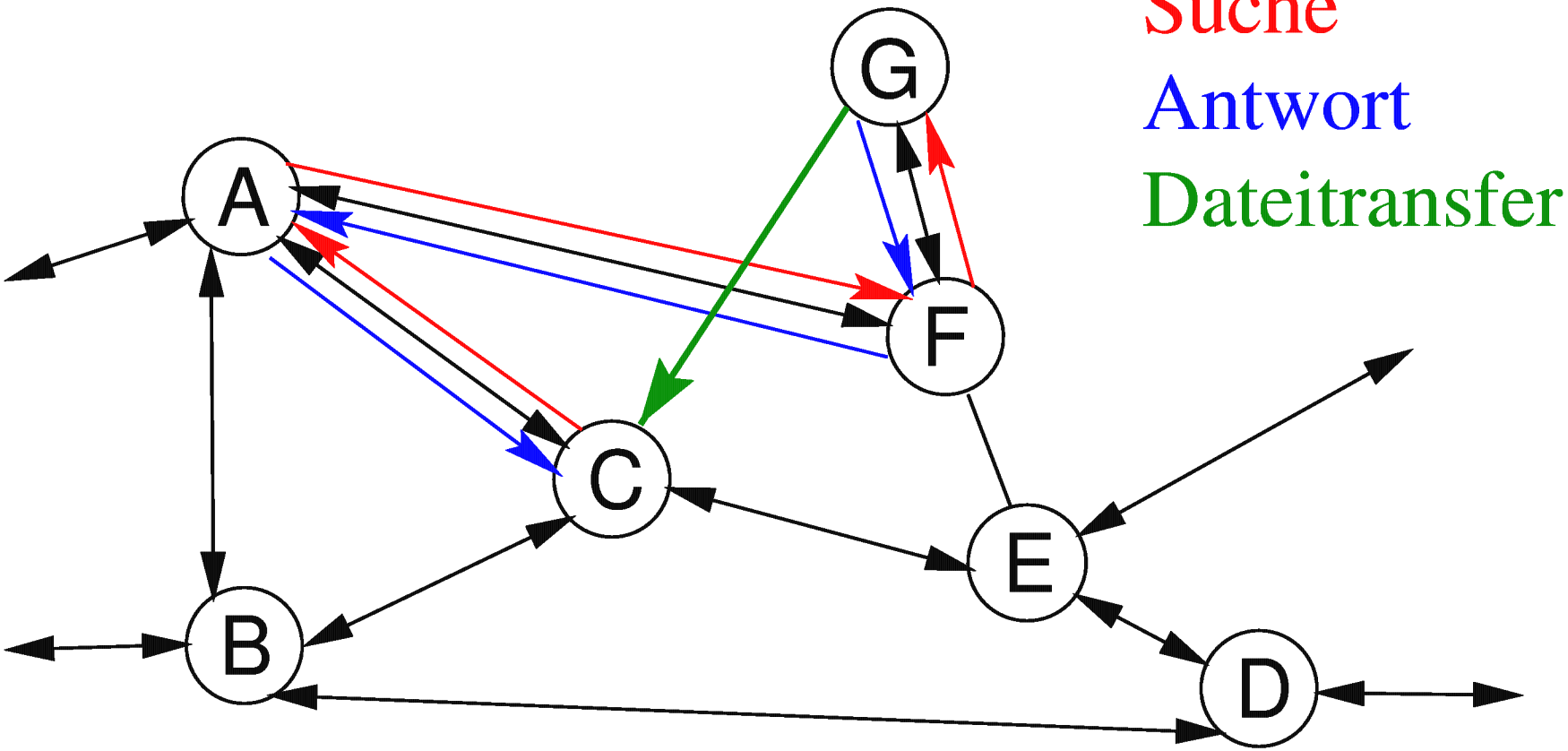
Suche von C aus, Treffer bei G



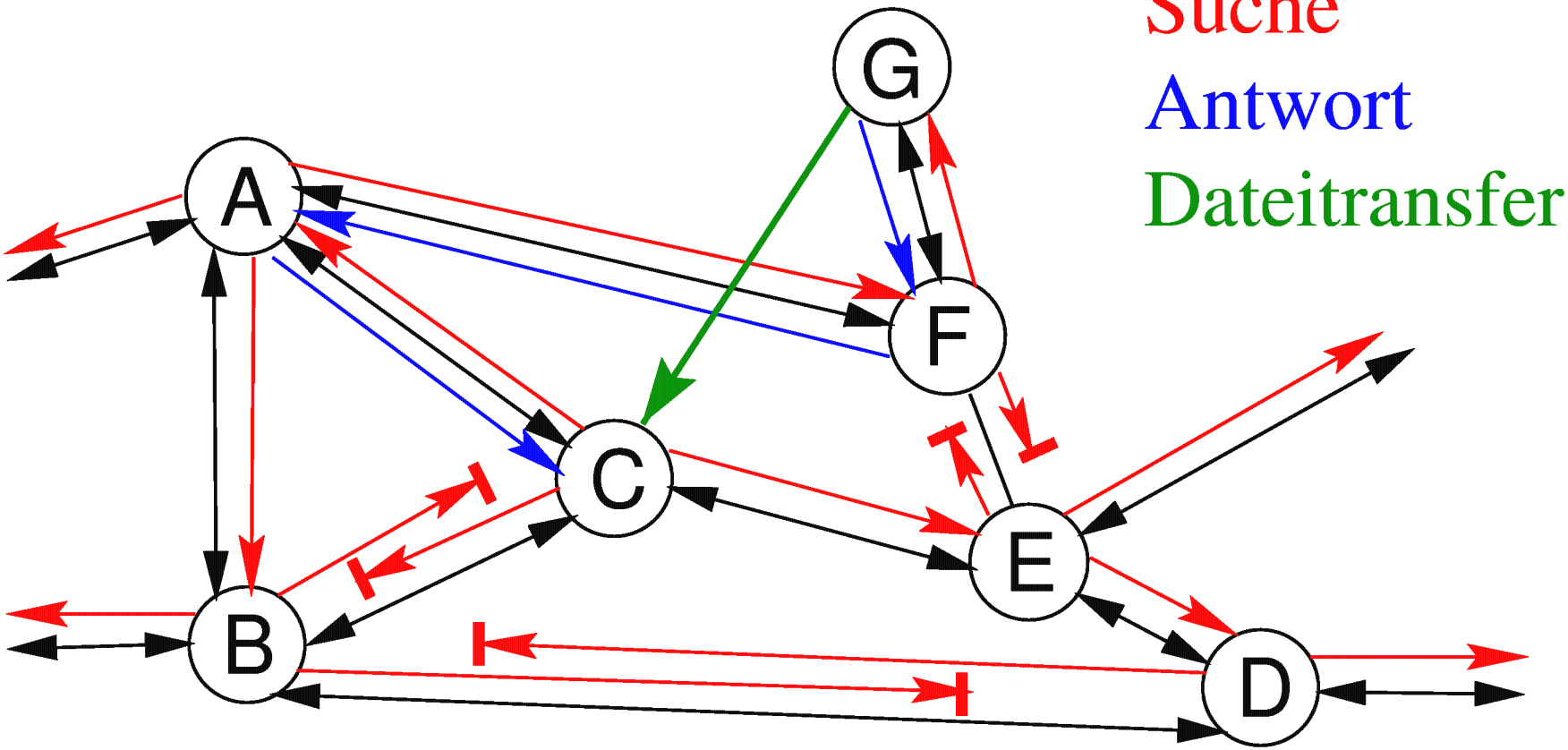
Suchantwort von G an C



Datentransfer von G direkt an C



Gesamtübersicht



Bemerkungen

- Chaotische Topologie: u.U. unzusammenhängend!
- Dynamsiches Verschwinden und Auftauchen von Knoten
- Suche ist tiefenbeschränkte Breitensuche
  - routing mit GUIDs
  - langsam
  - deckt u.U. nur Teilnetz ab
  - hohe Netzlast
  - Optimierung: Abweisen von Wiederholungen
- „Log On“ über Liste bekannter Knoten

Anonymität

- **Suche**
  - Anbieter: kennt Sucher nicht
  - Sucher: erhält IP-Adresse des Anbieters
- **Dateitransfer:**
  Direkter Kontakt, d.h. alles offen.
  $\Rightarrow$ Hall of Shame

Fazit Gnutella

- Keine zentrale Infrastruktur
- Durch Angriffe nur eingeschränkt störbar $\Rightarrow$ robust
- Zur systematischen Suche nur bedingt benutzbar
- Stand 6/2000: Verursacht starke Netzlast: ca. 2 kByte/sec pro direkte Verbindung bei ca. 5000 Knoten gleichzeitig aktiv
- Stand 11/2000: ca. 10.000-30.000 User pro Tag
  Kein Netzkollaps, aber starke fragmentierung
  Gute statistische Daten auf http://dss.clip2.com/gnutella.html

Beispiel III: Freenet

- **Ziel:** Unzensierbare Informationsinfrastruktur, vergleichbar WWW
- **Basis:** Diplomarbeit von Ian Clarke
- Informationen auf http://freenetproject.org
- Client = Server, Selbstorganisiertes Netz
- Entwicklung in JAVA
- **Status:** In Entwicklung, praktische Tests in Durchführung

Ziele von Freenet im Detail

- Anonymität für Publizierer
- Anonymität für Leser
- „Deniability“ für Speicherer
- Robustheit gegen Denial-Of-Service Attacken
- Routing und lokales Chaching effizient und bedarfsgesteuert
- Keinerlei zentrale Kontrolle

Topologie

Analog zu Gnutella, aber

- Daten werden geroutet
- Daten (und Datenquelle) werden in Durchgangsknoten in den lokalen Chache eingefügt
- Daten existieren **nur** in den Caches
- Bei Platzbedarf: erst Daten löschen, dann (später) Datenquellenzeiger löschen
  $\Rightarrow$ Wiederbeschaffungsmöglichkeit bei Anforderung

Suche

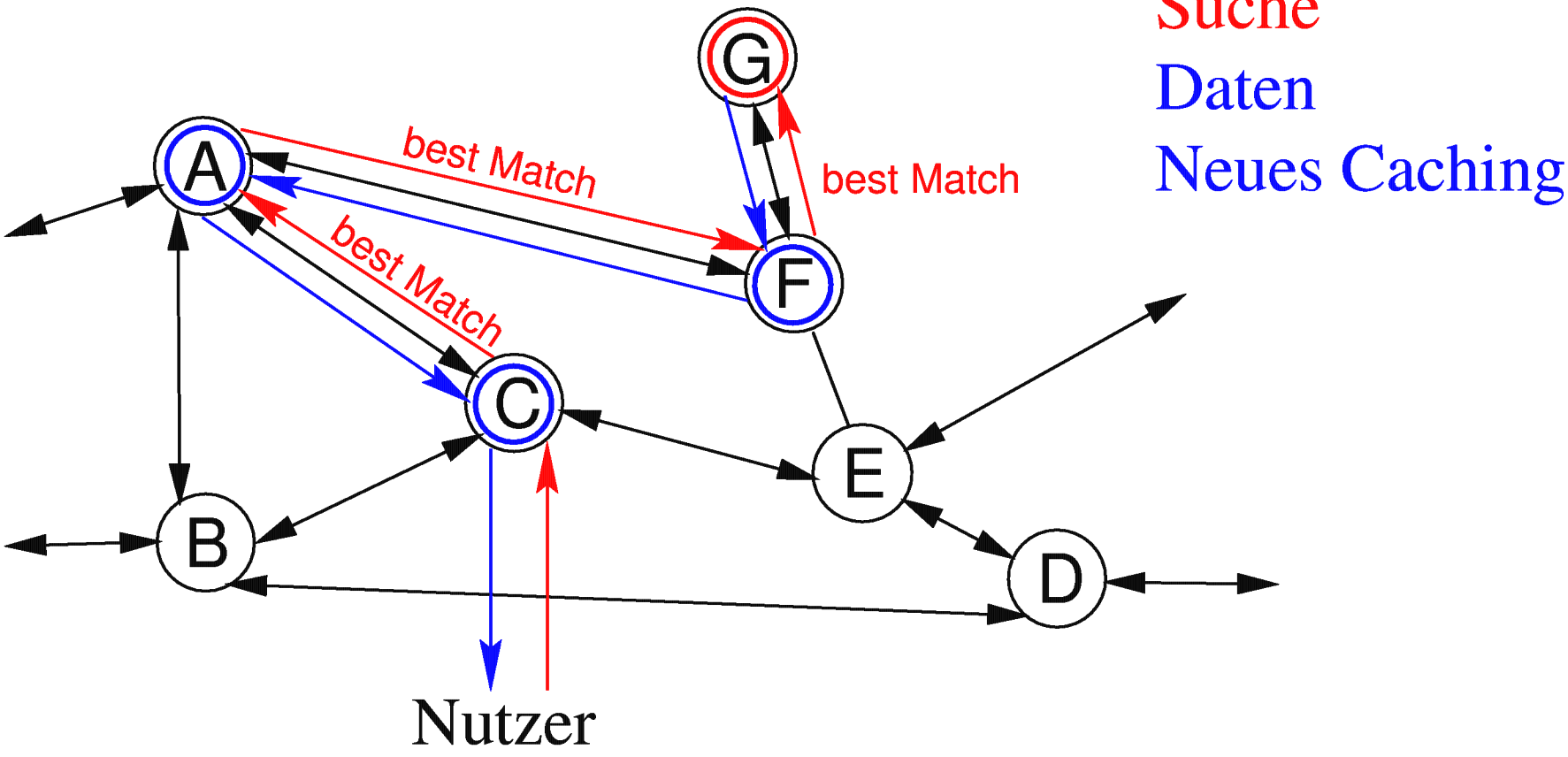
- Vollstringsuche („Key“)
- Keys sind Hashcodes
- Daten mit ähnlichem Key bilden Cluster
- Suchantwort: Daten! Keine explizite Anforderung!
- Suchmethode:
  - Lokale Suche nach ähnlichstem Key in Cache
  - Ähnlichster $\neq$ Gesuchter:
    Weitersuche in Herkunft des Ähnlichstem

  $\Rightarrow$ Selbstorganisation, Lastbalancierung

Problem: Formulierung von Suchanfragen

Lösung im Moment: Keylisten

Suche und Dateitransfer



Anonymität

- Gut, da
  - volles Routing (d.h. Fundort bleibt verdeckt)
  - keine Master-Kopien von Dokumenten vorhanden
- Kein Klartextzugriff $\Rightarrow$ Knotenbesitzer kontrolliert Inhalt nicht
  Aber: Problematisch mit Keylisten

Fazit zu Freenet

- Hohe Anonymität
- Leistung des Caching unklar
- Suchmechanismus unklar
- Datenmengenbelastbarkeit eher niedrig
- Suchzeiten unklar (gegenwärtig 1-60 sec)
- Funktioniert das Daten-Clustering bei grossen Netzen?
- Nichtentfernbarkeit wirklich wünschenswert?