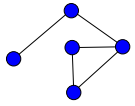
Gliederung

Peer-to-peer Systeme und Datenbanken(SS07)

- Kapitel 1: Einführung
- Kapitel 2: Beispiele
- Kapitel 3: Routing
- Kapitel 4: Schemabasierte p2p-Netzwerke
- Kapitel 5: Integrationsprobleme
  Teil 5-1: Einführung, Gleichheit
  Teil 5-2: Ähnlichkeit - 1
  Teil 5-3: Ähnlichkeit - 2
  Teil 5-4: Mappingbasierte Datenintegration
- Kapitel 6: Anonymität, Authentifikation
- Kapitel 7: Reputation

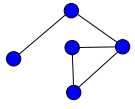
Version vom 4. Juli 2007



Kapitel 7 : Reputation

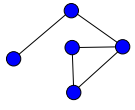
Gliederung:

- Motivation, Amazon, e-Bay
- Reputation. Was wird bewertet? Kosten der Bewertung
- Anforderungen
- Prizipielle Lösung
- Eigene Erfahrung vs. kollektive Meinung
- Verfahren mit mehreren Bewertungen
- Eigentrust-Verfahren



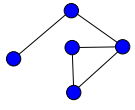
Motivation

- Free riding: eine egoistische Lösung des Kostenoptimierungsproblems. Widerspruch zur Grundidee der P2P-Netze, daß jeder Peer auch Serveranteile realisiert.
- Anonymität bewirkt, daß die Verantwortlichkeit für eingebrachte Inhalte verloren geht, das ist eine Einladung zum böartigen Mißbrauch.
  - Viren: z.B. *VBS:Gnutella*-worm, modifiziert Gnutella.ini
  - Bösartige Peers antworten auf alle Anfragen (spamming), geben unsinnige Antworten oder defekte/falsche Daten.
  - Gezielte Überlastung des Netzes oder einzelner Peers.
- Forderung: Trotz Anonymität und Fluktuation müssen Fehlverhalten erkannt werden: Identifikation und Isolation der betreffenden Knoten.



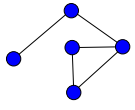
Bewertung bei Amazon

- erhöht die psychologische Hemmschwelle für Fehlverhalten
- Consumer product reviews:
  Potentielle Kunden können die Meinung von andern, die das Produkt bereits gekauft haben, lesen.
- Ansatz mit zentralem Server
- Für den Handel mit gewerblich hergestellten Produkten und mit kommerziellen Anbietern offenbar geeignet.
- Bewertung der Käufer? Bewertung der Verkäufer?



Bewertung by eBay

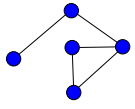
- Reputationssystem für Anbieter und Käufer
  Der Käufer wird hochpreisige Waren nur bei Verkäufern erwerben, die einen guten Ruf haben.
- Nach jedem Geschäftsvorgang kann der Käufer den Verkäufer bewerten und der Verkäufer den Käufer.
- Die Bewertung jedes Teilnehmers ist die Summe der Bewertungen der letzten 6 Monate.
- Zentrale Speicherung und Verwaltung
- Bewertung: Der zentralistische Aufbau vehindert einige Angriffsformen
  i.A. sehr gut: Anbieten schlechter Waren, Nichtbezahlung, ...
  Keine Aussage für Einzelfall.



Anforderungen / Ziel

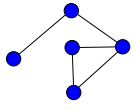
Kann eine Bewertung ähnlich der bei eBay in einem P2P-System ohne Zentralauthorität eingerichtet werden?

- Nur Nutzer, die sich regelgerecht verhalten, können die Dienste nutzen / auf Dauer nutzen.
- Ziel ist Finden und die Isolation eines auffälligen Peers, nicht die genaue Beschreibung oder Korrektur des Schadens, den er anrichtet.
- Nutzeridentifizierung und Anonymität scheinbar gegensätzlich.
  Wie kann Anonymität gewahrt werden?
- Ein wirksames Bewertungssystem wird wahrscheinlich selbst Gegenstand von Angriffen, muß also selbst Gegenstand einer Bewertung sein.



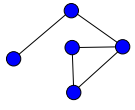
Anforderungen / Ziel (2)

- Im P2P-System müssen Bewertungsfunktionale integriert sein, so daß alle Aktionen der Peers hinsichtlich der Konformität mit den Zielen des Systems bewertetet.
  Funktional: Aktion $\mapsto$ (reelle) Zahl.
- Formen des Fehlverhaltens:
  Nutzung der Resourcen des P2P-Netzes ohne eigene Dienstangebote (Free-Riding)
  Aktives nicht regelgerechtes (bösartiges) Verhalten:
  - Falsche Beantwortung von Suchanfragen,
  - Verbreitung falscher Inhalte,
  - Falsche Bewertungen des Verhaltens der anderen Knoten.
- Reputation ist eine zusammenfassende Bewertung des Verhaltens eines Peers / der Korrektheit einer Resource auf Grund bisher durchgeführten Aktionen.



Reputation

- Grundidee aller Verfahren mit Reputation:
  Ein Peer zeigt immer das gleiche Verhaltensmuster, so daß aus der Reputation auf das Verhalten des Peers bei zukünftigen Aktionen geschlossen werden kann.
- Beispiel (Korrektheit von Daten): Ein Datensatz, der bei einem konkreten Peer liegt und der in der Vergangenheit bereits von mehreren anderen Peers als korrekt bewertet wurde, kann auch für weitere Verwendungen als (wahrscheinlich) korrekt angesehen werden.
- Reputation hinsichtlich verschiedener Kriterien möglich (s.o.)
  (Bewertungsvektor ?)
- Startproblem:
  Neuer Peer - keine (positive) Serverreputation - wird nicht als Quelle gewählt - ...
  (eine positive Startreputation wäre eine Angriffsquelle)

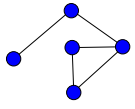


Vergleich mit Geld in Fragen

Idee des Vergleichs: Roger Dingledine u.a.: *Reputation in P2P Anonymity Systems*.
Mehrere Arbeiten zum Thema Anonymität in LNCS 2137,
URL `http:www.freehaven.net/papers.html`

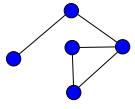
- Wie bekommt man Reputation? Transitivität?
- Vergeht Reputation mit der Zeit?
- Gibt es eine ständige Quelle, ein Reservoir? Gibt es Kredite?
  Oder entsteht R. nur beim Systemstart und wie?
- Ist R. global oder hat jeder Knoten seine lokale Bewertung, wie reagiert dann das System auf Fluktuation?
- Kann eine globale Bewertung aus den lokalen R. entstehen?
  Umrechnungen?

Antworten nicht eindeutig, variieren nach Anforderungen $\mapsto$ Lösungsvielfalt.



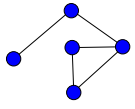
Beispiel: Remailer-Reputation

- Ziel: Anonynes Versenden von Nachrichten durch Bündelung und Senden durch Ketten von Remailern, dadurch wird die einzelne Nachricht nicht-verfolgbar. Kein Glied der Kette darf versagen, Zusammenstellung der Ketten nach der Reputation der Glieder.
- Bewertung sowohl gutes Verhalten als auch Fehlverhalten bewerten, andernfalls Angriffsmöglichkeiten
  nur posititiv - Startproblem
  nur negativ - Angreifer erhält durch gr. Zahl eigener Nachrichten gute Bewertung
- Zusätzliche Nachrichten zum Aufbau der Bewertung: Rückmeldeung in der Kette, Sender kann Verbleib der Nachricht prüfen, Mitteilungen und Anfragen an globale Zeugen (*weakly trusted global witnesses*), Testnachrichten.
  Witnesses dürfen nicht manipulierbar sein, sind evt. Ziel von DoS-Angriffen.
- in diesem System: Reputation - in langem Prozeß erworben, keine Sicherheit für Zukunft, keine Sicherheit im Einzelfall.



Was wird bewertet?

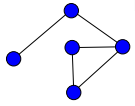
- Peer muß sowohl als Server als auch als Client Verhaltensnormen erfüllen, Konkretisierung abhängig von Systemzielen.
- In einem file-sharing System Client: Download/Upload-Verhalten
  Server: Qualität der Antworten (Suchanfragen)
  Server/Provider: Qualität der angebotenen Daten
- Die Qualität der Bewertungen muß selbst bewertet werden, andernfalls eröffnet die Bewertung selbst neue Angriffsmöglichkeiten.
  (Eigensicherheit der Bewertung)
- Möglichst eine skalare Größe zur Bewertung, um eine Vergleichbarkeit zu erhalten.
- Grundsätzliche Entscheidung:
  - Bewertung nur nach eigener Beobachtung
  - Austausch der Bewertung mit Nachbarknoten
  (Abwägung Vorteile vs. Risiken)



Beispiele bösartigen Verhaltens

- Einbringen nicht korrekter Informationen (defekte Dateien - teilweise als Mittel zur Realisierung der Urheberrechte eingesetzt, Viren)
  $\mapsto$ inhaltliche Bewertung der gelieferten Resultate
- Positive Beantwortung von Suchanfragen, ohne daß die entsprechende Information geliefert werden kann (Nichtlieferung oder Lieferung falscher Informationen - kann erst nach der Lieferung bewertet werden).
  Kann zur Informationssammlung dienen (Brechung der Anonymität, wird als Mittel zur Ermittlung von Urheberrechtsverletzung genutzt).
- Weiterverbreitung von Informationen, die falsch sind
  (fahrlässig, falls keine Korrektheitsprüfung, obwohl möglich).
- Nichtbeantworten von Suchanfragen, Nichtweiterleiten von Nachrichten, Vernichten von Informationen.
  Schwer zu trennen von Fehlern auf Ebene des Internet.
- Falsche Bewertungen.

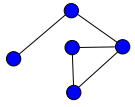
Hier nicht erfaßt: DoS und andere nicht spezifische Angriffe.



Bösartiges Verhalten hinsichtlich der Bewertung

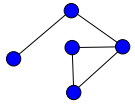
Lit.:Marti und Garcia-Moulina:*Limited Reputation Sharing in P2P Systems.*
EC04, May 2004, New York, USA.
Annahme: Ziel ist Verbreitung falscher Inhalte.
Klassifikation hinsichtlich der Bewertungsvarianten:

- Normalverhalten: keine Fehlinformartion, die verbreitete Meinung über andere ist korrekt.
- Isolierte Lügner: Fehlinformationen mit den Ziel eigener Vorteile oder falsche Meinungen über alle anderen Knoten verbreitet.
- Kooperierende Lügner: Verbreiten gute Meinungen über Mitglieder der Kooperationsgruppe und schlechte über korrekt arbeitende Knoten.
  Frontknoten: Spez. Mitglieder einer Gruppe koop. L., verteilen nur korrekte Inhalte, um gute Bewertungen zu erhalten und dann ihre Fehlbewertungen anderer Knoten (erfolgreicher) zu verteilen.
  shilling: Gruppe tauscht falsche Information unter sich, um Bewertung zu manipulieren.



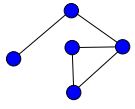
Kostenfaktoren

- Jede Bewertung setzt eine Qualitätsprüfung voraus, verursacht damit Kosten.
  Maß für die Güte einer Reputation: wieviele Bewertungen sind nötig, bis ein korrektes Ergebnis des Service festgestellt wird (Erwartungswert)
- Bestandteile der Kosten:
  - Soll die Qualität eines Service beurteilt werden, muß er zuvor in Anspruch genommen werden (Servicekosten)
  - Kosten für Nutzung des unter dem P2P-Netz transparenten Internet (z.B. Tauschbörsen: Zeit/Volumen zum Download)
  - eigentliche Qualitätsbewertung der Ergebnisse
- Qualitätsbewertung häufig nicht automatisierbar.
  (Tauschbörsen: Software muß installiert und getestet werden, selbst dann ist die korrekte Funktion fraglich; Musik muß vollständig durch geschulte Personen abgehört werden; Rechenaufträge ???)
  $\mapsto$ hoher personeller Einsatz



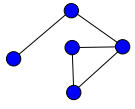
Design-Kriterien

- Selbst-konfigurierend: die Nutzergemeinschaft gibt sich selbst die Regeln vor, keine zentrale Administration.
- Anonymität: die Bewertung eines Peers soll an *systeminterne* Identifikatoren gebunden werden, die so anonym sind wie auch das gesamte betrachtete P2P-System.
- Keine Bonusvorschuß für Newcomer: Reputation muß durch regelgerechtes Verhalten und Mitarbeit im System erworben werden (ein bösartiger Peer würde sich sonst immer wieder neu anmelden).
- Minimaler Zusatzaufwand hinsichtlich Rechenkapazität, Infrastruktur, Speicherplatz, Nachrichtenanzahl.
- Robustheit gegenüber kollektiven Angriffen: Eine Gruppe bösartiger Peers versucht in koordinierten Aktionen des Bewertungssystem zu umgehen oder wirkungslos zu machen.



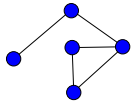
Knoten-Identifikation

- Jede Bewertung ist notwendigerweise an (untäuschbare) Identifikation der Knoten gekoppelt.
- Strenge Anforderungen $\mapsto$ vertrauenswürdiger Server mit login-Verfahren, Peer-ID von realer Welt eindeutig abgeleitet, kann Anonynität wahren (Hash), kann aber nicht gewechselt werden. Problem: Dynamische IP-Adressen
- Schwache Identitätsanforderungen: Nutzergenerierte Identität
  Angriffe gegen Bewertung:
  - Identitätswechsel,
  - Benutzung mehrerer Pseudonyme durch einen Peer (die dann wie ein Angreiferkollektiv wirken),
  - Benutzen von Identitäten wohlreputierter Knoten (gestohlene I.).
  - Man-in-the-middle Angriff



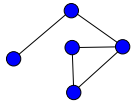
Prinzipielles Vorgehen

- n Knoten, jeder Knoten führt Bewertungsvektor über alle anderen Knoten $\mapsto n \times n - Reputationsmatrix R$: $R_{i,j}$ Meinung von i über j, (Bespielsemantik: größerer Wert = bessere Bewertung).
- Verfahren zur Dienstauswahl: Knoten q fragt Service an, erhält Angebote durch verschiedene andere Knoten.
  q bewertet jedes Element der Antwortmenge mit der Reputation des die Antwort gebenden Knotens.
  (*) Auswahl eines Angebots
  Inanspruchnahme des Angebots - Bewertung der Resultate - Aktualisierung des lokalen Bewerungsvektors
  falls die Bewertung der Resultate nicht positiv ausfällt ist bei (*) fortzufahren.
- Variationsmöglichkeiten:
  - Variationen der Auswahlkriterien.
  - Nur Benutzung der lokalen Bewertung oder der Bewertungsvektoren anderer Knoten.



Auswahlkriterien

- Auswahl des Kandidaten mit der besten / höchsten Bewertung:
  erfordert Speicherung einer lokale Liste einer Länge l der besten bisher bekannten Knoten.
  Kritik: evt. Konzentration auf wenige gutbewertete Knoten und Integrationsproblem für neue Knoten.
  Variante: (*friends-first*-Technik) Anstatt Anfrage an alle Knoten zu richten, wird sie nur an die Mitglieder der o.g. Liste gesendet (Netzlast $\downarrow$), nur wenn diese eine leere Antwortmenge liefern, ergeht eine allgem. Anfrage.
- Gewichtete Auswahl: Aus der Anwortmenge wird zufällig ein Knoten gwählt, die Wahrscheinlichkeit, daß q den Knoten j wählt, ist proportional $R_{q,j}$.
  Ziel: bessere Lastverteilung der Anfragen.



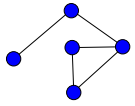
Auswahlkriterien (2)

Einbeziehung der Wertung anderer Knoten:

- Knoten q fragt Service nach und erhält Antwortmenge A.
  q hat eine Menge Q von Knoten, deren Meinung ihm wichtig ist.
- Für jede Antwort aus A, die von einem Knoten r stammt, werden alle Knoten $v \in Q$ um ihre Meinung $R_{v,r}$ gefragt
  $\mapsto$ Gesamtwertung ρ_r für Knoten r bei Knoten q:

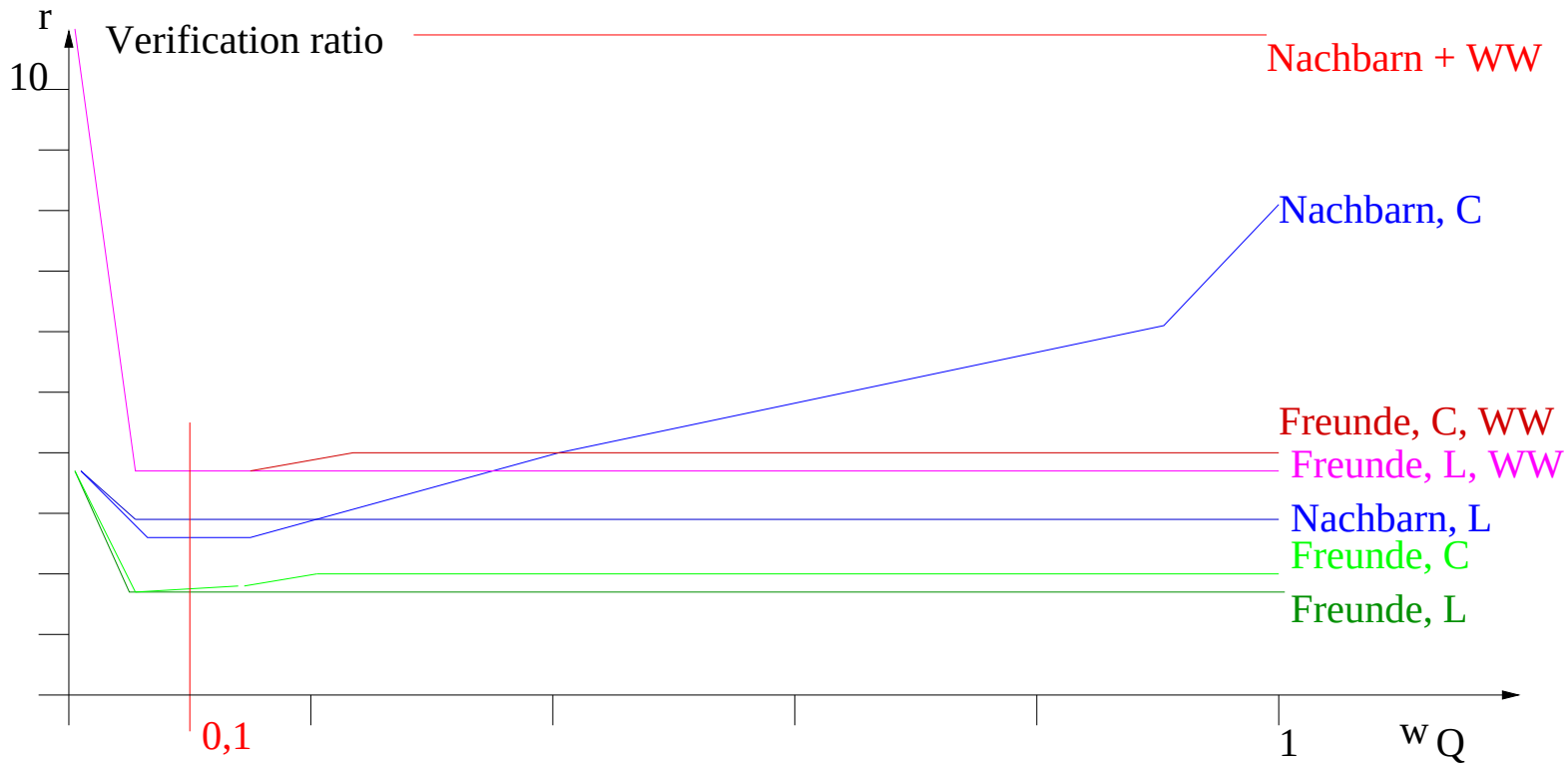
$$\rho_r = (1 - w_Q) R_{q,r} + w_Q \frac{\sum_{v \in Q} R_{q,v} R_{v,r}}{\sum_{v \in Q} R_{q,v}} \qquad (1)$$

  Lokale Wertung ist Spezialfall bei $w_Q = 0$.
- Auswahl der Mitglieder von Q:
  - Nachbarn entsprechend der Netztopologie
  - Knoten, die bisher hoch bewertet wurden (*Freunde*).
- Auch periodischer Abgleich der Wertungen möglich (Netzlast).



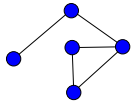
Simulationen (Marti, Garcia-Moulina)

Effizienz: Anzahl der Verifikationen pro korrektes Resultat



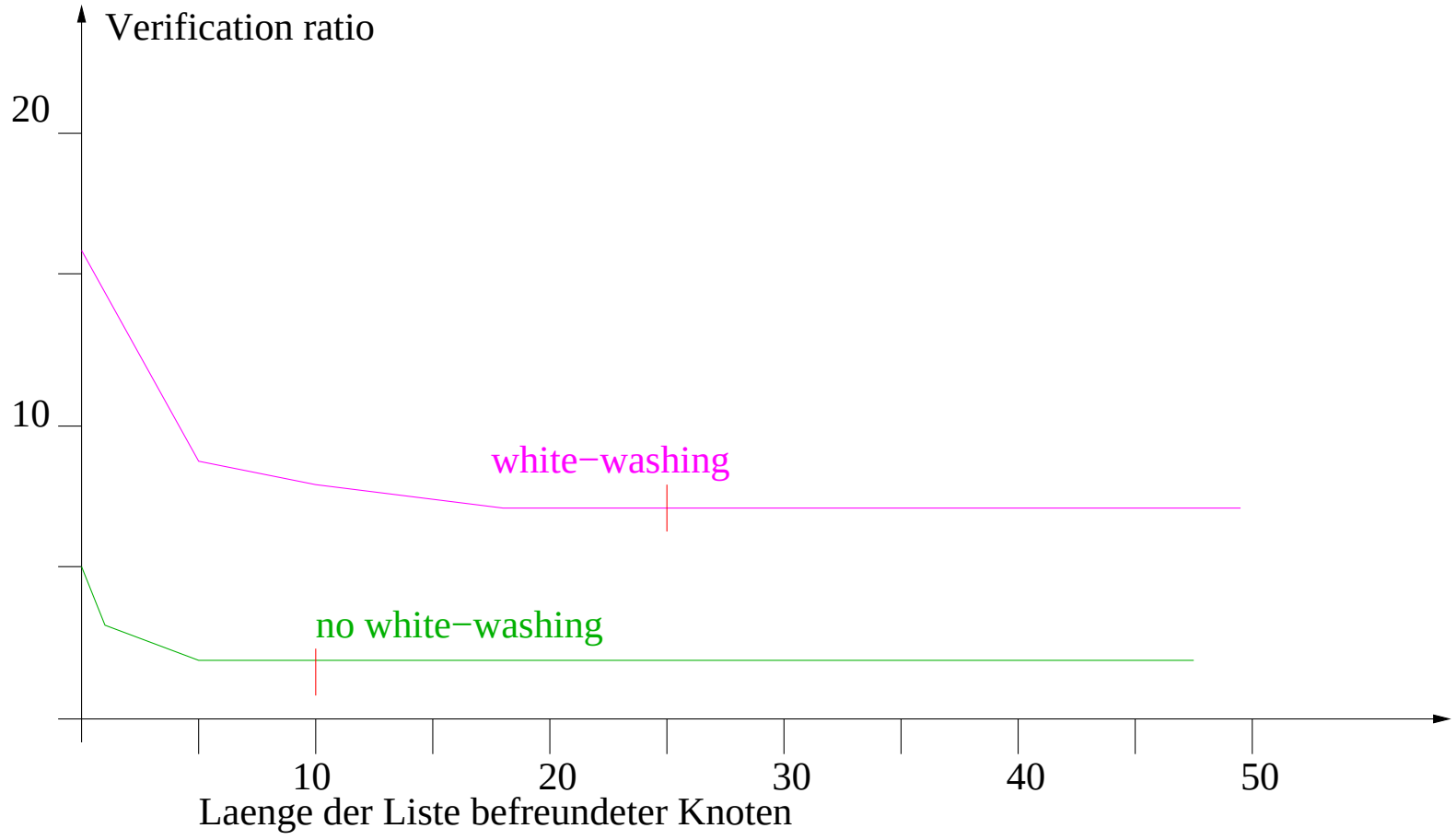
WW Whitewashing
L egoist. Einzellügner; C koop. Lügnergruppe

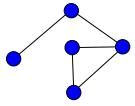
Alle Kurven stark vereinfacht.



Simulationen (2)

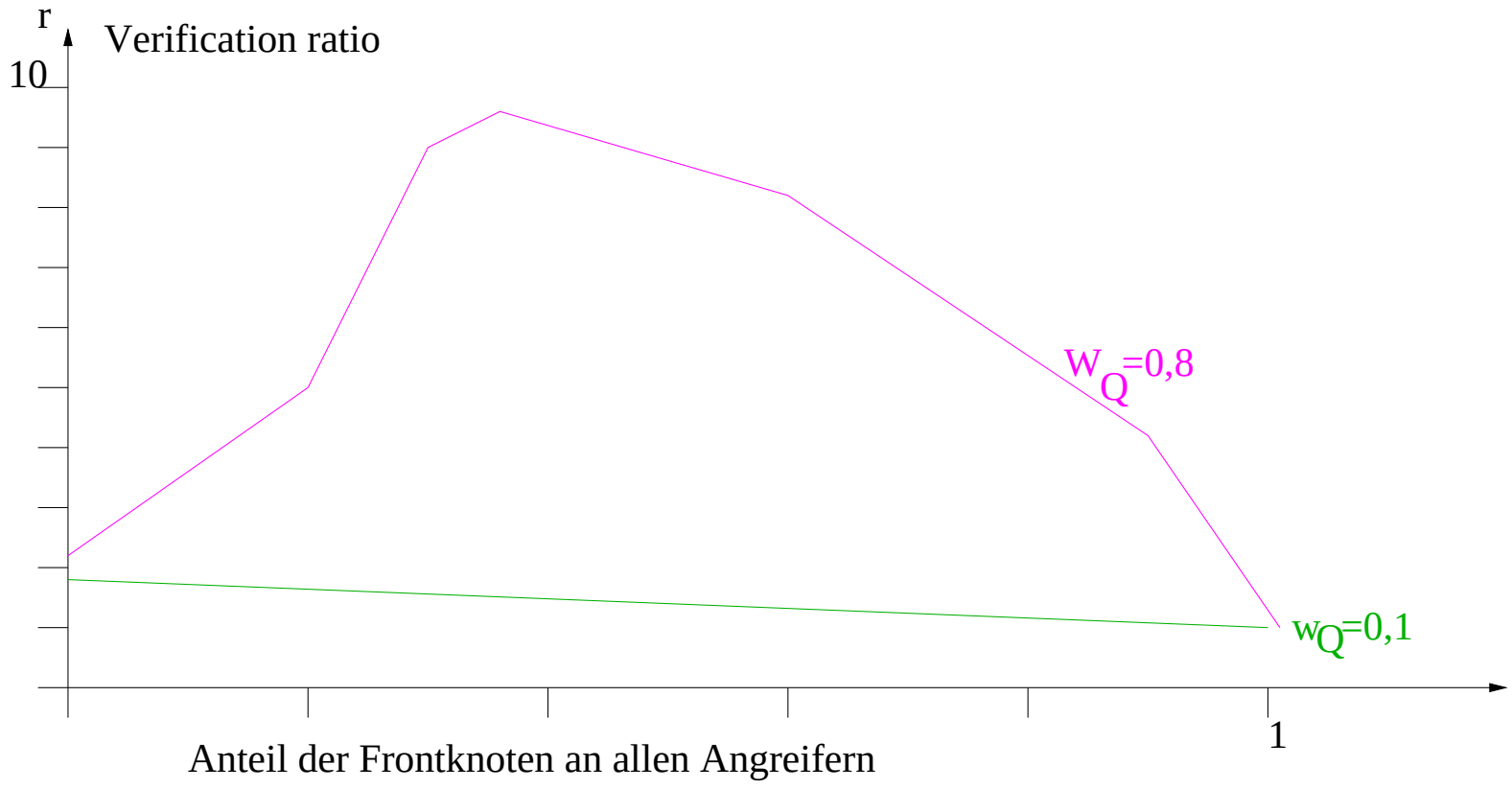
Effizienz in Abhängigkeit von der Anzahl befreundeter Knoten.

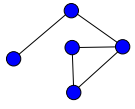




Simulationen (3)

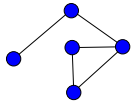
Einfluß von Frontknoten





Simulation (4) - Weitere Ergebnisse

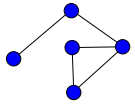
- Lastverteilung bei Auswahl des besten Knotens:
  Spitzenwert: 10-fache Last des Durchschnitts
  Lastverteilung bei gewichteter Auswahl: Spitzenlast halbiert.
- bei Auswahl des besten Knotens werden Knoten bevorzugt, die viele Daten anbieten.



Kombinierte Verfahren

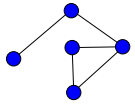
Lit: Makoto Iguchi u.a.: *Managing Resource and Servent Reputation in P2P Networks*. Proc. 37^{st} Hawaii International Conference on System Sciences - 2004

- Kombination von Bewertung der Qualität der Resourcen und der Arbeit der Knoten der P2P-Netzes.
- Bewertung der Qualität jedes Datum r mit *recource reputation score* R.
- Bewertung Arbeit der Server-Knoten s als Quelle von Inhalten und als Bewerter.
  Reputationsbasiert mit 2 Bewertungen:
  servent contribution score SC und *servent evaluation score* SE
- Jeder Knoten i speichert lokal Bewertungen aller Knoten j, von denen er jemals Daten angefordert hatte (diese Informationen ggf. auch weitergeben). Bewertungen dienen zur Auswahl der Partner für weitere Informationsanforderungen.
- Nach jeder Transaktion werden die Qualität bewertet und die Bewertungen aktualisiert.



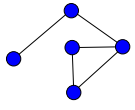
Beschreibungen der Bewertungen

- *recource reputation score* R_l: Maß für Vertrauenswürdigkeit der Resource r_l, summiert die bisher erhaltenen Bewertungen für r_l.
  Ein hoher Wert bedeutet, daß r_l gute / positive Bewertungen erhielt.
- *servent contribution score* SC_i bewertet das bisherige Verhalten des Knoten s_i als Verteiler von Information.
  Ein hoher Wert bedeutet, daß die Daten, die s_i verteilt hat, von anderen Peers positiv (als korrekt) eingeschätzt wurden.
- *servent evaluation score* SE_i bewertet, ob die vom Knoten s_i abgegebenen Bewertungen von Resourcen von anderen Peers bestätigt werden.
  Ein hoher Wert bedeutet, daß abgegebene Bewertung als korrekt eingestuft wurde.
- Die 3 Bewertungen werden separat gehalten.



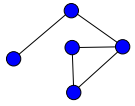
Grundannahmen A

1. Eine Resource ist vertrauenswürdig, wenn der Peer, der sie anbietet, selbst als Quelle vertrauenswürdig ist.
2. Eine Resource ist vertrauenswürdig, wenn ein Peer, der bisher Resourcen korrekt bewertet hat, sie anbietet.
   (Bemerkung: *korrekt* bedeutet hier *korrekt / richtig*.
3. Ein Server ist als Quelle vertrauenswürdig, wenn die Resourcen, die er in der Vergangenheit bereitgestellt hat, gute / positive Bewertungen erhielten.
4. Ein Server ist als Bewerter vertrauenswürdig, wenn die Resourcen, die er positiv (negativ) bewertet hat, auch von anderen positive (negative) Bewertungen erhalten.



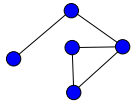
Regeln zur Berechnung der Bewertung

- (Annahme A1) $\mapsto$ Die Recource-Reputation R_l einer neuen Resource r_l, die von Server s_i eingebracht wird, ergibt sich aus dem Contribution-Score SE_i von s_i.
- (Annahme A2) $\mapsto$ Die Resource-Reputation R_l einer existierenden Information wächst (fällt), wenn diese Recource positive (negative) Bewertungen von einem Server s_j erhält, für den die Einschätzung seiner Bewertungen (SE_j) positiv ist. Diese Einschätzung wird auch quantitativ ausgewertet. Ist jedoch SE_j negativ, kehrt sich die Richtung des Zuwachses von R_l um.
- (Annahme A3) $\mapsto$ SC_i eines Servers s_i wächst (fällt) wie die Resource-Reputation derjenigen Informationen, die er bereitstellt, im P2P-Netz wächst (fällt).
- (Annahme A4) $\mapsto$ SE_i des Servers s_i wächst (fällt) wie die Resource-Reputation R_l einer Recource r_l, wenn die Richtung seines Votums mit der Änderungsrichtung von R_l übereinstimmt (nicht übereinstimmt).



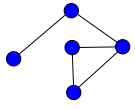
Implementierung im hybriden System

- Annahmen: Zentraler Server, der die Wertungen verwaltet, eindeutige Identifizierbarkeit von Daten und Peers.
- Protokoll:
  Anmeldung mit Übergabe der Resource-Liste an Server,
  Anfragen an Server, Antwort v. Server,
  Auswahl d. Quelle aus Antwortmenge,
  direkte Informationsübertragung,
  Bewertung u. Aktualisierung der Resource-Reputation und der Server-Reputationen



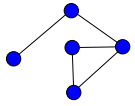
Hybride Systeme (2)

- Falls Server s_i beim Anmelden Resourcen r_l anbietet, für die Zentrale keine Reputation hat, setzt sie nach A1:
$$R_l = SE_i$$
- Antwort der Zentrale auf Suchanfrage:
answer := (resource-info, servent-info +)*
resource-info := { resource-id l, r_l, resource-decription }
servent-info := { servent-id i, SC_i }+
- Auswahl: Resource mit maximalem R,
falls die Bewertungen alle (zu) niedrig sind, wähle Anbieter mit maximalem SE.
- Bewertung (mit Werten aus dem Intervall $[-1, 1]$.
Bewertungsnachricht an Zentrale: (servent-id, resource-id, votum)



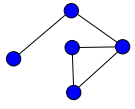
Bewertungsberechnung im zentralen Server

- Resource-Reputation:
  Server s_i mit Evaluationsbewertung SE_i hat Datum r_l mit v_{li} bewertet, entsprechend A2 wird gerechnet:
  $\Delta R_l = v_{li} \times SE_i$,
  $R_l := R_l + \Delta R_l$
- Zentrale ändert SE desjenigen Knoten S_j, der r_l angeboten hat entsprechend A3: $SE_j := SE_j + \Delta R_l$
- Die Änderung des Resource-Reputation-Score von r_l hat Folgen fr alle *anderen* Knoten, die r_l jemals bewertet hatten wegen A4: Wenn die Knoten s_k, $k = 1, ..., n$ für r_l die Voten v_{lk} abgegeben haben, ergibt sich ihre neue Evaluationsbewertung:
  $SE_k := SE_k + v_{lk} \times \Delta R_l$, $k = 1, ..., n$
- Alle *weiteren* Knoten s_k, die r_l jemals mit einer Wertung v_{lk} bewertet haben, erhalten wegen A4 eine Änderung ihres SE_k:
  $SE_k := SE_k + v_{lk} \times \Delta R_l$, $k = 1, ..., n$
  (Bemerkung: Produkt hat stets das richtige Vorzeichen.)



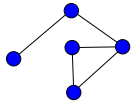
Resistenz

- Identitätswechsel: Ein Server mit negativer/n Reputation/en meldet sich mit neuer Identität an.
  Abwehr: Ein neuer Peer erhält eine schlechtere Start-Bewertung, z.B die schlechteste jemals vergebene Bewertung.
  Peer muß sich eine gute Bewertung erarbeiten:
  Bereitstellung von Daten, die schon eine gute Resource-Reputation, haben
  $\mapsto$ Aufbau eines SC
  Korrekte Bewertung der genutzten Resources $\mapsto$ Aufbau des SE.
- Manipulation der Serverreputation: Kein Server ist allein in der Lage, seine Reputationen zu verändern, In seine Wertung gehen nur die Voten anderer ein (Zeugenfunktion) $\mapsto$ zu einem physischen Knoten darf es keine Pseudonyme geben.
- shilling-Angriffe (nächste Folie)



Resistenz gegen shilling

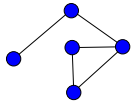
- (Shilling: die Mitglieder einer Angreifergruppe tauschen fehlerhafte Informationen unter sich, um die Bewertung zu manipulieren.)
- Angreifer beim shilling sind keine Pseudonym-Knoten.
  Shilling in dem vorgestellten Modell hat nur gute Wirkung, wenn die Falschbewertung von Knoten i kommt, die einen hohen SE haben:
  $\Delta R_l = v_{li} \times SE_i$,
  Diese Bewertung SE_i mußte zuvor erarbeitet werden.
  Es ist unwahrscheinlich, daß ein Knoten mit guter Reputation (SE) shilling betreibt, da dadurch sein SE sinkt:
  $SE_k := SE_k + v_{lk} \times \Delta R_l$, $k = 1, ..., n$
  Noch unwahrscheinlicher ist das Zusammengehen vieler gutbewerteter Server, die Angreifer müssen jedoch einen deutlichen Anteil am Netz haben, insbesondere wenn nicht nur sie die Resource anbieten.



Implementierung als reines P2P-System

Wesentliche Änderungen:

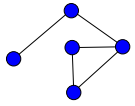
- Jeder Server s_i hat lokale Reputationsdatenbank:
  $\{(r_l, R_l)\}$ aller Resourcen r_l, die er jemals hatte,
  $\{(s_j, SC_j, SE_j)\}$ aller Server, von denen er jemals Daten erhielt,
  $\{(v_{l,j}, s_j\}$ mit jedem Votum, das s_i jemals gesendet hat, wenn er r_l geholt hat und jedem Votum, was er von anderen Servern auf seine Anforderungen hin erhalten hat.
- Austausch /Abgleich mit anderen Servern (da kein Weltwissen existiert)
  Modifikation der einzelnen Schritte:
  Ergebnis einer Anfrage: Menge $\{(r_l, s_j)\}$
  Anfrager muß selbst eine Bewertung der Ergebnisse durchführen, ehe eine Auswahl getroffen werden kann. Grundsätze bleiben.



Dezentrale Bewertung von r_l bei s_i

adaptiert die Regeln der Version mit zentralem Server

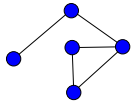
- Falls R_l lokal vorhanden, dann $R_{l:temp} = R_l$
  R_l ex. nicht lokal, aber SC_j des Anbieters - nach A1: $R_{l:temp} = SC_j$
  Wenn beides nicht lokal existiert: Anlegen eines neuen Servereintrags in der DB mit den üblichen (schlechten) Startwerten für SC und SE.
- Für jede r_l Anfrage an die anderen Server nach Voten über r_l.
  Antwortmenge für jede r_l: $\{(v_{lk}, s_k)\}$
  Neuberechnung der Resource-Reputationen R_l wie bekannt mit folgender Modifikation:
  Wenn SE_k bei s_i bekannt ist, dann diesen Wert verwenden, andernfalls s_k bisher unbekannt, wird er vor der Rechnung als neuer Knoten (s.o.) eingeführt.
  (Dublikate entfernt, neuester Wert verwendet.)
- Quellenauswahl wie im zentralen Fall.



Aktualisierung der Datenbank

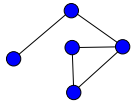
Nach Erhalt von r_l gibt der Empfänger eine Bewertung v ab
$\mapsto$ Akualisierung der lokalen Datenbank:

- Resource-Bewertung: $\Delta R_l = v$; $R_l := R_l + \Delta R_l$.
- SC_j des gewählten Anbieters s_j:
  Mit dem so berechneten R_l bilde $\Delta_1 R_l = R_l - R_{1:temp}$,
  $SC_j := SC_j + \Delta_1 R_l$
- SE_k aller Knoten, die ihr Votum zu r_l abgegeben haben:
  $\Delta_2 R_l := R_l - R_{l:temp} - v_{lk}$;
  $SE_k := SE_k + \Delta_1 R_l$
  (d.h. das Votum von s_k geht in seine eigene Bewertung nicht ein.)



Probleme dieses Ansatzes

- Angreifer mit gefälschter Adresse (man-in-the-middle Angriff) kann die lokalen Bewertungen der echten Knoten durch falsche Voten absenken $\mapsto$ Störung der Auswahl und der weiteren Reputation.
- Angreifer, der zu einer Resource-ID r_l eine gefälschte Resource kann.
- „self-centered lokal reputation“: Lokale Datenbank mit begrenztem Horizont
  Ein Knoten s_i kann einen anderen mittelmäßigen Knoten s_j als sehr gut einschätzen, weil er auf Grund eines Gesichtsfeldes über diesen Knoten nur sehr gute Meinungen erhalten hat (Schwäche gegenüber zentr. Bewertung)
  Lösung: „reputation equalization“- regelmäßiger Austausch der lokalen Bewertungen SC und SE mit anderen Knoten und Anpassung der eigenen Bewertung, wenn große Differenzen auftreten.



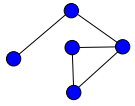
Eigentrust Algorithm

Literatur: Kamvar, S.D. u.a.: *The EigenTrust Algorithm for Reputation Management in P2P Networks*

- Mathematisch begründetes Modell zur Lï£¡ung der Frage:
  Wie kann in einem reinen P2P-System aus lokalen Bewertungen eine globale Bewertung errechnet werden ?
  Resistenz gegen Angriffe?
- lokaler Vertrauensvektor bei Peer i: $s_{i,j} = \sum_k tr_{i,j,k}$,
  $tr_{i,j,k} = 1$ bei pos. bewerteter Transaktion k zw. i und j (aus Sicht von i)
  $tr_{i,j,k} = -1$ bei neg. Bewertung der T. k aus Sicht von i.
  Normierung:

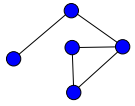
$$c_{i,j} = \begin{cases} \max(s_{ij}, 0) / \sum_j \max(s_{ij}, 0) & \text{falls } \sum_j \max(s_{ij}, 0) \neq 0, \\ \text{undefiniert}, & \text{falls } \sum_j \max(s_{ij}, 0) = 0 \end{cases}$$

  (Anmerkung: nach Normierung keine Unterscheidung zw. Peers, mit denen i schlechte Erfahrungen bzw. keine Interaktion hatte.)



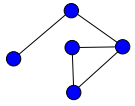
Eigentrust - Matrix

- Übertragung der Bewertung an dritten Peer:
  i fragt seine Nachbarn j, was sie über k denken (c_{jk})
  und summiert die Meinungen (gewichtet mit eigener Meinung über j :
  $t_{ik} = \sum_j c_{ij} c_{jk}$
- Matrixschreibweise: $C = \{c_{ij}\}, \vec{c}_i = \{c_{ij}\}$, j-te Spalte der Matrix
  $\vec{t}_i = C^T \vec{c}_i$
- i will die Meinung der Nachbarn seiner Nachbarn einbeziehen:
  $\vec{t}_i = (C^T)^2 \vec{c}_i$
- Trustvektor: Unter (einigen math.) Voraussetzungen konvergiert das Verfahren bei weiterer Wiederholung unabhängig, mit welchen Werten (an welchem Knoten) es gestartet wurde (gegen den Eigenvektor zum größten Eigenwert von C).
  Näherung: $\vec{t} = (C^T)^n \vec{e}, n$ hinr. groß.



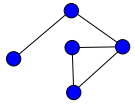
Einfaches lokales Verfahren

- Startwerte $\vec{e}$ nicht relevant $\mapsto$
  $\vec{e}$ entweder alle Komp. gleich 1/m , m Anzahl der Peers
  oder lokaler Vektor $\vec{c}_i$ oder ...
- Iterationsverfahren konvergiert: ε gegeben.
  $\vec{t}^{(0)} = \vec{e}$;
  repeat ($\vec{t}^{(k+1)} = C^T \vec{t}^{(k)}$; $\delta = \|\vec{t}^{(k+1)} - \vec{t}^{(k)}\|$;)
  until $\delta < \varepsilon$
- Verbesserungen: a-priori-Vertrauen, inaktive Peers, bösartige Gruppen
  (1) Ein a-priori-Vertrauen $\vec{p}_j$ für einige Peers j bekannt: wird als $\vec{e}$ eingesetzt, andere Peers dort Null.
  Bessere Konvergenz des Verfahrens mit diesem Startwert, falls es bösartige Peers im System gibt.
  (2) inaktive Peers: Peer i hat selbst nichts geholt oder hat zu niemanden Vertrauen $\mapsto \sum_j \max(S_{ij}, 0) = 0$ und c_{ij} nicht definiert:
  setze für dieses i: $c_{ij} = p_j, j = 1, ..., m$.



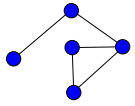
Bösartige Gruppen

- Eine bösartige Gruppe in einem P2P-System mit Reputationsverwaltung ist eine Gruppe von Peers, die
  - einander kennen
  - den Gruppenmitgliedern ein hohes Vertrauen schenken und anderen Peers ein geringes,
  - um selbst eine gute Reputation zu bekommen und die korrekte Funktion des Systems zu stören.
- Hat sich eine solche Gruppe erfolgreich etabliert und kontaktiert ein Außenstehender ein Gruppenmitglied, wird er in der Gruppe gehalten.
- Gegenmaßnahme: $\vec{t}^{(k+1)} = (1-a)C^T\vec{t}^k + a\vec{p}$
  $\vec{p}$ ... Vektor des a-priory-Vertrauens $\mapsto$ Iteration modifiziert.
- Kein Peer mit a-priory-Vertrauen darf Mitglied in einer bösartigen Gruppe sein.
  Auswahl: Nach Kriterien außerhalb des Bewertungssystems, z.B. Initiatoren eines P2P-Systems, Teilnehmer, die keinen Eigennutz aus dem System ziehen (können).



Verteilte Berechnung

- Alle Peers eines Netzwerks arbeiten bei der Berechnung zusammen, Zusatzaufwand für Berechnung, Speicherung und Nachrichtenaustausch bei jedem Peeer mï£¡lichst gering.
- Jeder Peer i speichert seinen lokalen Vektor $\vec{c}_i$ und seinen globalen Vektor $\vec{t}_i$.
  (Hinweis: Bei der Implementierung ist zu beachten, daß dann kein Peer seine eigene Bewertung manipulieren darf!)
- Globale Grundformel:
  $\vec{t}^{(k+1)} = (1-a)C^T\vec{t}^k + a\vec{p}$
- Lokal(beim i-ten Peer): $t_i^{(k+1)} = (1-a)\sum_{l=1}^{m}(c_{li}t_l^k) + ap_i$
  Da i nicht alle anderen Peers kennt, bleiben viele Koeffizieten Null.

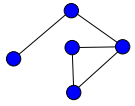


Verteilte Berechnung (2)

- Jeder Peer i rechnet wie folgt:
  A_i : Menge der Peers, die von i Daten abgerufen haben;
  B_i : Menge der P., die an i Daten geliefert haben.

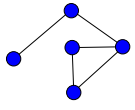
Anfrage alle $j \epsilon A_i$ nach $t_j^{(0)} = p_j$;
repeat
  $t^{(k+1)_i} = (1-a) \sum_{l=1}^{m} (c_{li} t_l^k) + a\vec{p_i}$;
  sende $c_{ij} t_j^{(k+1)}$ an alle Peers $j \epsilon B_i$;
  $\delta = |t^{(k+1)_i} - t_i^k|$;
  erwarte von Peers $j \epsilon B_i$ die Werte $c_{ji} t_j^{(k+1)}$
until $\delta < \varepsilon$

- Wichtig für Sicherheit / Anonymität der Peer mit a-priori-Vertrauen: nur sie kennen ein p_i (nur das eigene!).
  Zunächst vorausgesetzt: alle betrügen nicht.



Aufwand

- Jeder Knoten in einem großen Netz hat nur (relativ) wenige andere Peers, mit denen er Transaktionen hatte. Damit sind viele c_{ji} gleich Null und die Berechnung von $t_i^{(k+1)} = (1-a)\sum_{l=1}^{m}(c_{li}t_l^k) + a\vec{p_i}$ erfordert nur geringen Aufwand.
- Die Mengen A_i, B_i enthalten (relativ) wenige Elemente $\mapsto$ Geringe Zahl von Nachrichten.
- Simulationen: 1000 Peers , k=10 Iterationen.
  $\|t^{(k+1)} - t^k\|_1 \approx 0.01$
- Simulationen hat gezeigt: Peer muß nicht alle lokalen Vertrauenswerte bei Abfrage senden, dennoch konvergiert das Verfahren.



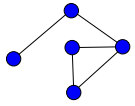
Sicherheit

Bisher vorausgesetzt: Kein Peer betrügt bei der Rechnung. Bösartige Peers?
Neue Ideen zur Verbesserung der Sicherheit

- Kein Peer berechnet seine eigenen Werte. Jeder Wert wird mehrfach berechnet und durch Mehrheitsentscheid angenommen.
- Jeder Peer i hat M Peers als *score manager*, diese werden mit verteilten Hash-Tabellen (CAN, CHORD) angeordnet. Manager speichern die Werte der Peers in ihrem Gebiet. Aus der ID eines Peers können seine Manager bestimmt werden. Manager übergeben ihre Daten bein Verlassen des Netzes.

Bezeichnungen: D_i Menge der Peers, für die i Manager ist,
für jedes $d \in D_i$ ist c_d^i der lokale Vetrauensvektor von d, verwaltet von i
B_d^i Menge der Peers , von denen $d \in D$ Daten Resourcen nutzte.

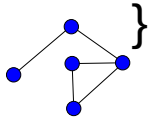
A_d^i Menge der Peers, die von d Resourcen nutzten



Sicheres Verfahren

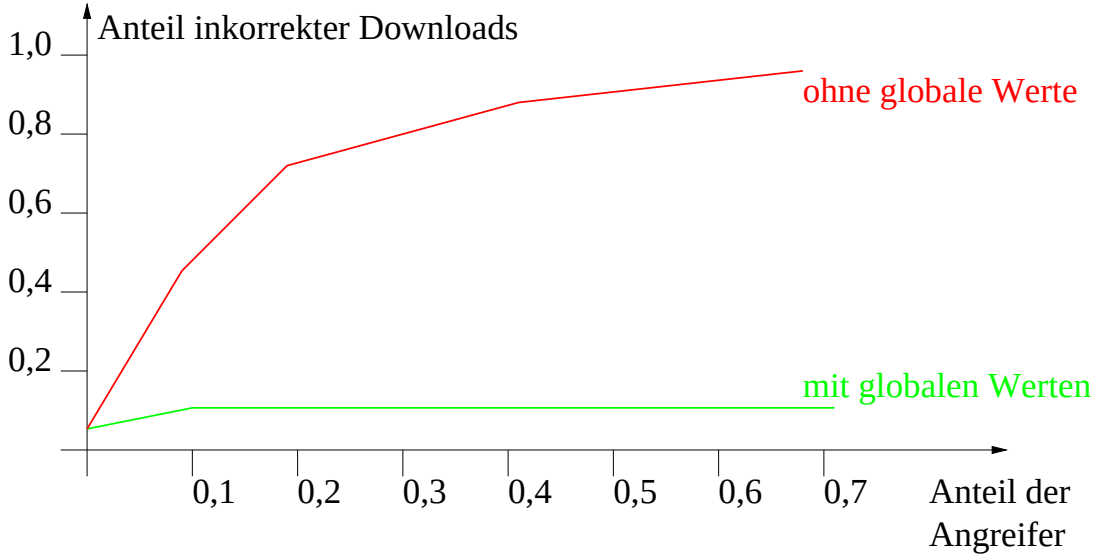
Algorithmus: Kamvar u.a., a.a.O.

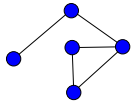
foreach peer **as** i {
  submit local trust values $\vec{c}_i$ to score managers of i;
  collect $\vec{c}_d$ und B_d^j von $d \in D_i$;
  foreach $d \in D$ {
    submit c_{dj} an die anderen Manager von d;
    ask $j \in A_d^i$ for $c_{jd}p_j$;
    repeat
      $t_d^{(k+1)} = (1-a)\sum_{l=1}(c_{ld}t_l^k) + a\vec{p_d}$
      submit $c_{dj}t_d^{(k+1)}$ an alle $j \in B_d^i$
      waitfor $c_{jd}t_j^{(k+1)}$ von allen $j \in A_i^d$;
    until $|t_d^{(k+1)} - t_d^k| \leq \varepsilon$
  }
}



Wertung

- Vorteile:
  Anonymität: kein Peer weiß, wessen Reputation er berechnet, Angreifer können andere Angreifer nicht bevorteilen.
  Zufälligkeit: Hash - Einwegfunktion, Peer kann seine ID nicht so wählen, daß er in ein gewünschtes Gebiet kommt.
  Redundanz: Peer hat mehrere Scoremanager.
- Wirkung

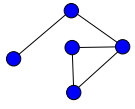




Anwendungen

Die Bewertungen können mit verschiedenen Zielen erfolgen:

- Free-Rider werden Anbieten von Daten angeregt
- Es ist attraktiv, viel gesuchte Daten anzubieten.
- Erkennen von nicht korrekten Resourcen
- Isolation von bösartigen Peers (einzeln, Gruppen)
- Lastverteilung



Zusammanfassung

- Bewertung des Verhaltens der Knoten ist wichtig für korrekte Funktion des Systems.
- Bewertung muß hinsichtlich verschiedener Kriterien erfolgen:
  Resourcen hinsichtlich der Korrektheit
  Peers hinsichtlich des Einbringens und Verteilens korrekter Resourcens und hinsichtlich korrekter Bewertungen der Resourcen und der anderen Peers.
- Bewertung bedeutet auch Identifikation, Kompromiß mit Anonynitätsforderung.
- Bewertung, hauptsächlich die inhaltliche, verursacht Kosten.
- Berechnung globaler Bewertungen auch in reinen P2P-Systemen möglich.
- Wirksamkeit in Simulationen nachgewiesen, es gibt kaum Resultate von Messungen an realen Systemen.