

Übung zu Peer-to-Peer und Cloud Computing

Übungstermin 7: Besprechung des Übungsblattes 6

Dominik Rauh

19. Dezember 2018

Universität Augsburg

Institut für Informatik

Lehrstuhl für Organic Computing

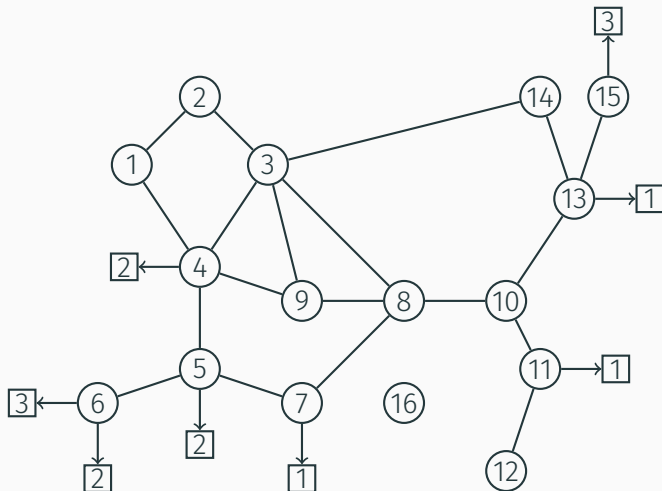
Dieses Mal wurden 20 Punkte zum Bestehen benötigt (entspricht 68 %).

Komplexität unstrukturierter P2P-Netzwerke

Schätzen Sie die Komplexität für das Einfügen,
Löschen und Suchen von Dateien in einem
unstrukturierten Netz ab.

- Einfügen
 - $O(1)$
 - keine Kommunikation erforderlich
 - Datei wird einfach lokal „veröffentlicht“
- Löschen
 - $O(1)$
 - keine Kommunikation erforderlich
 - Datei wird einfach lokal gelöscht
 - wenn alle Kopien gelöscht werden sollen: $O(n)$ für n Knoten
- Suchen
 - $O(n)$
 - im schlimmsten Fall: alle Knoten durchsuchen (z. B. wenn gesuchtes Objekt nicht vorhanden)

Unstrukturierte P2P-Netzwerke: Rechenaufgaben



Ausgehend von Knoten v_1 wird die Ressource r_3 gesucht.

Gegeben sind drei Suchverfahren:

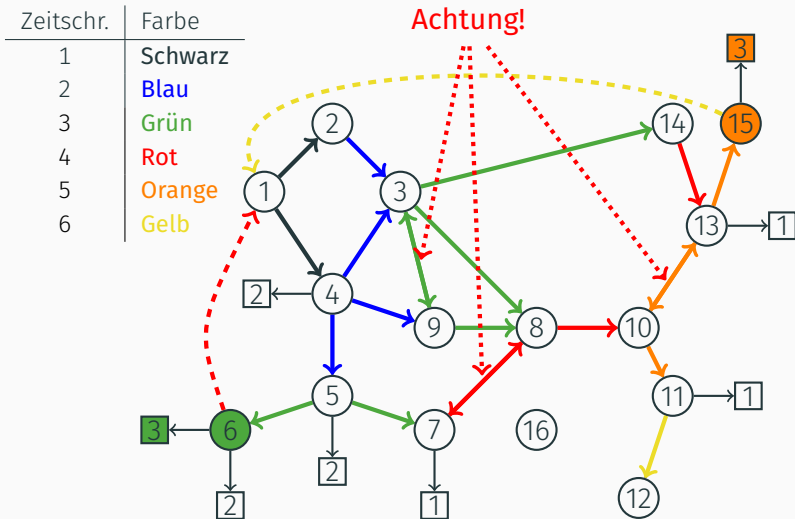
1. *Flooding* (uneingeschränkt)
2. *Expanding Ring Search* mit $TTL \in \{1, 2, 3, 4, 5\}$
3. Tiefensuche

1. Alle gesendeten Nachrichten einzeichnen!
2. Wie viele Nachrichten werden insgesamt verschickt?
3. Wie hoch ist die Latenz bis zur Antwort?

In P2P-Netzwerken werden Antworten immer möglichst *direkt* geschickt!

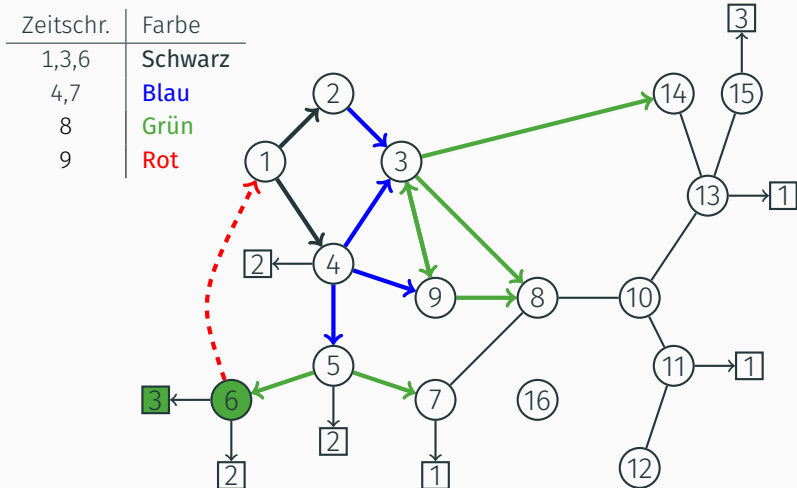
- Hauptgrund: Performanz
- Warum auch nicht: Das Overlay ist nur ein *virtuelles* Netzwerk!
- in strukturierten P2P-Netzwerken war es auch schon so (z. B. in Chord-Ringen)
- in unstrukturierten sollte es noch klarer sein:
Es gibt sowieso (fast) keine Regeln für den Aufbau von Verbindungen!

Schwarze
 Blau
 Grün
 Rot
 Orange
 Gelb

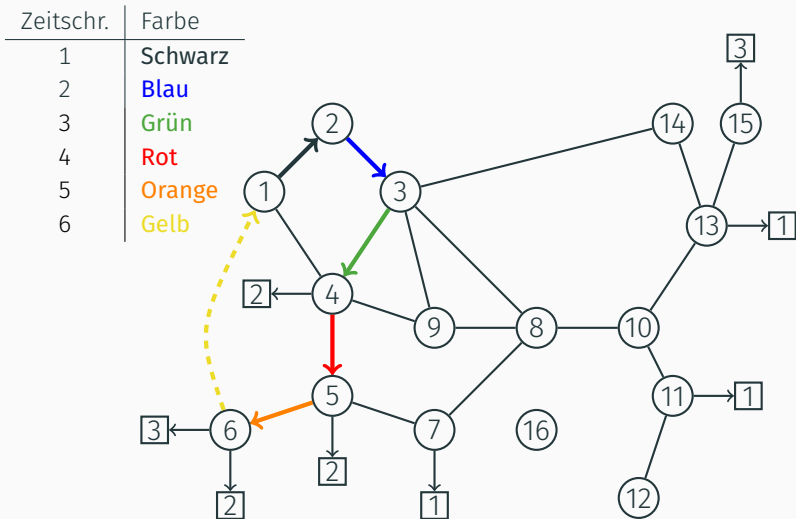


- 24 Nachrichten insgesamt, davon zwei Antwortnachrichten
- Antwort kommt im vierten Zeitschritt bei v_1 an

- 22 Nachrichten insgesamt, davon eine Antwortnachricht
- Antwort kommt im neunten Zeitschritt bei v_1 an (zwischen den einzelnen Flooding's jeweils einen Zeitschritt auf eine mögliche Antwort warten)



- Flooding mit $TTL = i$ in Zeitschritt i
⇒ viel mehr Nachrichten (fast unbeschränktes Fluten)
- neues Flooding startet, wenn vorheriges gerade $TTL = 0$ erreicht
⇒ in diesem Fall bis $TTL = 4$



- 6 Nachrichten insgesamt, davon eine Antwortnachricht
- Antwort kommt im sechsten Zeitschritt bei v_1 an

Wählen Sie eine *sinnvolle* Heuristik und beschreiben Sie kurz ihren Ansatz.

lokaler Index mit Radius 2

Beantworten Sie die Fragen aus Aufgabe 1 für
Ihre Art der Breitensuche.

Farbe

Schwarz
Blau
Grün
Rot
Orange

- 16 Nachrichten insgesamt, davon vier Antwortnachrichten
- (erste) Antwort nach zwei Zeitschritten

Oder natürlich ein lokaler Index mit Radius 3 ...