

Architektur verteilter Anwendungen

- Schwerpunkt: verteilte Algorithmen
- Ideen und Konzepte für algorithmische Probleme in verteilten Systemen
- Praktische Aufgaben: Implementierung dieser Konzepte zur Erkennung der Problematik

BYOD = bring your own device or use ISL

Architektur verteilter Anwendungen

- Vorlesung mit 5 integrierten Projektaufgaben
Lehrform eher 1V+1S+2P
- Anmeldung per e-Mail: Name+Matnr.
- Webseite für Veranstaltung
www-crypto.htwsaar.de/weber/teaching/14_ws_ava/
- Übungsblätter, Status der Bewertung,
Abnahmeplan

Architektur verteilter Anwendungen

- Prüfungsleistung:
 - praktische Aufgaben (50%)
 - mündliche Prüfung (50%)
- Bonuspunkte für die mündliche Prüfung
 - 30 min Vortrag eigener Lösung: 10%
 - max 3 eigene Vorträge (Dienstag)

ISL 5103

- Treffpunkt Mi/3. für Fragen, Betreuung, Abnahme STL-Loginaccount via ssh, STL-Homepartition C/C++/Python/Ruby, Java auf isl-s-02
- isl-s-01
Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz
- isl-s-02
Quad-Core AMD Opteron(tm) Processor 2350
- isl-s-04
Intel(R) Xeon(R) CPU E5440 @ 2.83GHz

Projektaufgaben

- Übungen müssen vor Abgabe besprochen sein (Abnahme)
- Übungen enthalten ein README (.txt, .doc, .odt)
 - Erläuterung der Idee
 - Nachrichtenformat
 - Erläuterung der Softwarestruktur
 - Hinweise auf Implementierungsbesonderheiten
 - typische Beispielabläufe
 - Fazit (gewonnene Erkenntnisse)
- Programmcode kommentiert

Letzte Abnahmemöglichkeit: Mittwoch 28.01.2014

Abnahmeplan

- Anmeldung zur Abnahme Mi (später auch Di) per e-Mail
(bis 18:00 Uhr am Vortag)
- aktualisierter Abnahmeplan auf AVA-Seite
- ca. 6 Abnahmen / Mittwochsdoublestunde

Programmcode

- lernen Sie eine neue Sprache
 - Python # aber Kommentare benutzen
 - Ruby # keine Scheu vor Anmerkungen
 - C# // kommentieren wie in Java
 - ...
- übersichtlich, modular, selbsterklärend

Bewertung der Abgabe

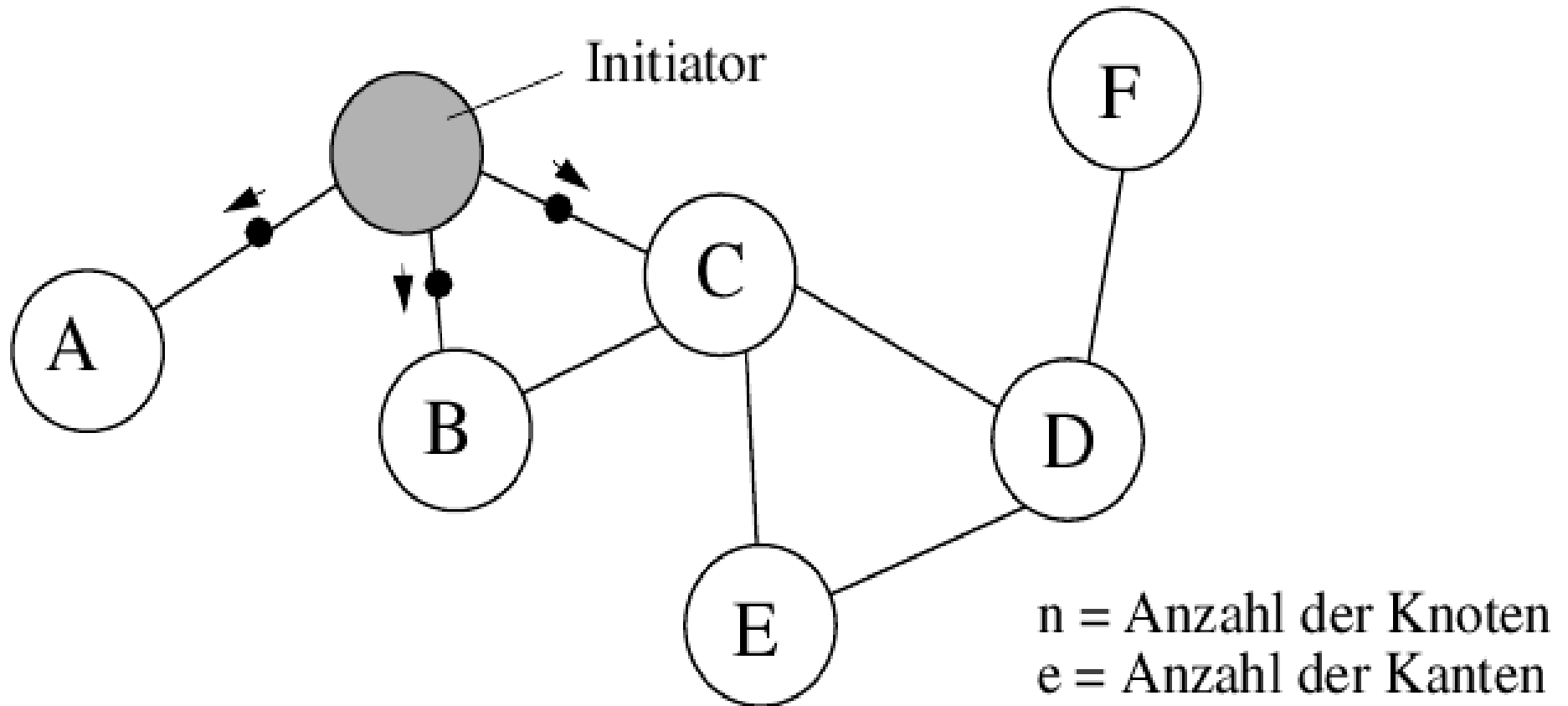
- 5 Punkte: Übung ist komplett, kommentiert, korrekt, skalierbar
- 4 Punkte: skaliert nicht oder kleinere Mängel
- 3 Punkte: korrekt für einen Spezialfall
- 2 Punkte: nicht besprochen, schwierig zu testen, Teile fehlen
- 1 Punkt, 0 Punkte: mangelhafte Umsetzung

Motivation



Einer hat die Info, jeder hat Freunde, Verbreitung eines Gerüchts.

Abstraktion



Knoten = lokale Algorithmen Kanten=Verbindungen zum Nachrichtenaustausch

Algorithmus

```
R: {Eine Nachricht <info> kommt an}
  if not informed then
    send <info> to all other neighbors;
    informed := true;
  fi
```

```
I: {not informed}
  send <info> to all neighbors;
  informed := true;
```

Schwierigere Aufgabenstellung



Jeder habe eine Karte mit einer Zahl, gesucht ist die größte Zahl.

Noch schwieriger ...



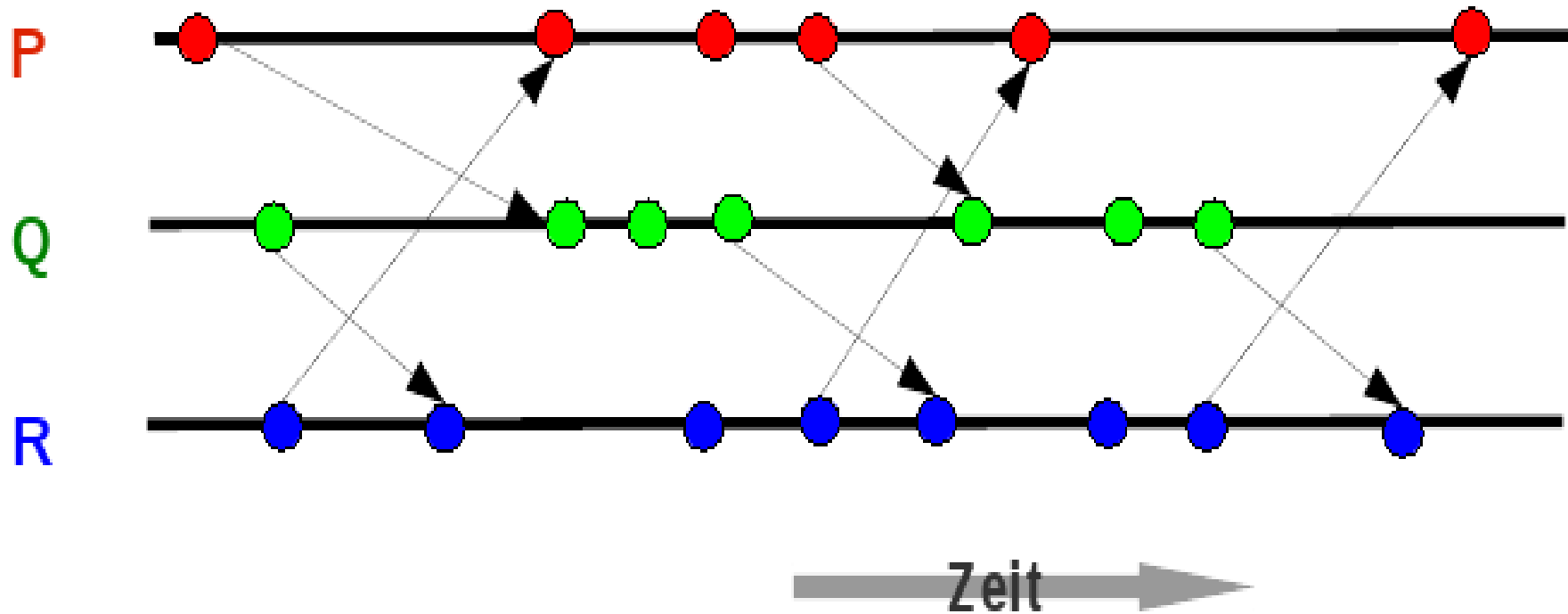
Jeder hat Bargeld. Jeder schenkt jedem ständig Geld.

Wieviel Geld insgesamt im Umlauf?

Verteilte Algorithmen

- Algorithmen symmetrisch
 - gleichberechtigt
 - kein Client-/Server, sondern autonome Knoten
 - Atommodell:
 - Rechnen in Nullzeit
 - Nachrichten mit endlicher Laufzeit
 - Nachrichten können sich nicht überholen (FIFO)
- asynchrones Senden bzw. Empfangen

Raum-Zeit-Diagramme



Modelle für verteilte Anwendungen

- sehr individuell, Kurse heißen
 - Verteilte Systeme
 - Verteilte Anwendungen
 - Verteilte Algorithmen
 - Nicht-sequentielle Programme
 - ...

Übung 1

- Basisalgorithmus: autonomer Knoten
 - für das gesamte Semester benutzt
 - empfangen Nachricht und sende an Nachbarn
- Erweiterung des Basisalgorithmus
 - graphviz
 - Topologie ausrechnen