

# Die Komplexität des Zuverlässigkeitsproblems in verteilten Systemen

Robert Siegfried

Hochschule Mittweida (FH)  
Mathematisches Seminar

20. Januar 2005

# Agenda

Grundlagen

Komplexität des Zuverlässigkeitsproblems

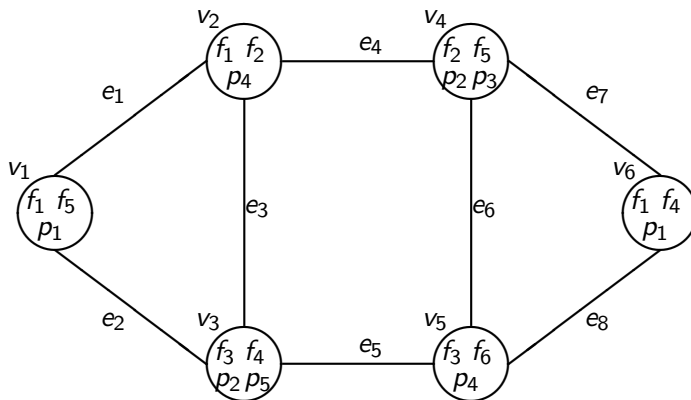
Fazit

# Verteilte Systeme

Ein *verteiltes System* (engl. Distributed Computing System, DCS) besteht aus:

- ▶ Verarbeitungselementen (Knoten)
- ▶ Kommunikationsverbindungen (Kanten)
- ▶ Daten/Dateien
- ▶ Programmen
- ▶ Speicher

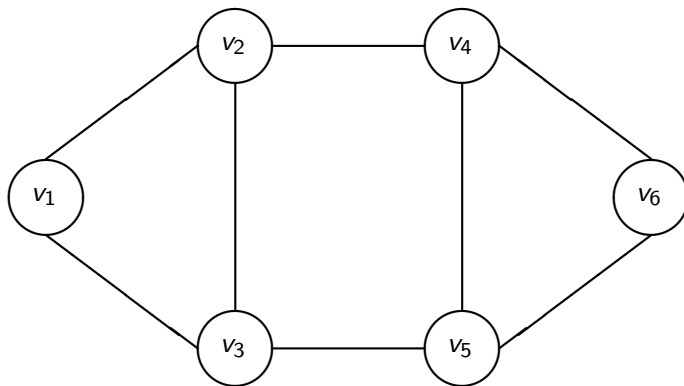
## Beispiel für ein verteiltes System



## Zuverlässigkeitsmaße

- ▶ Die *K-Terminal-Reliability* (KTR) ist die Wahrscheinlichkeit mit der eine gegebene Menge von Knoten  $K \subseteq V$  eines Graphen verbunden bleibt, wobei die Kanten unabhängig voneinander mit bekannten Wahrscheinlichkeiten ausfallen können.
- ▶ Die *Distributed Program Reliability* (DPR) ist die Wahrscheinlichkeit, dass ein Programm mit verteilten Dateien erfolgreich verarbeitet wird, unter der Annahme, dass die Knoten eine perfekte Zuverlässigkeit besitzen und die Kanten unabhängig voneinander mit bekannten Wahrscheinlichkeiten ausfallen können.

## Beispiel für ein verteiltes System (KTR)



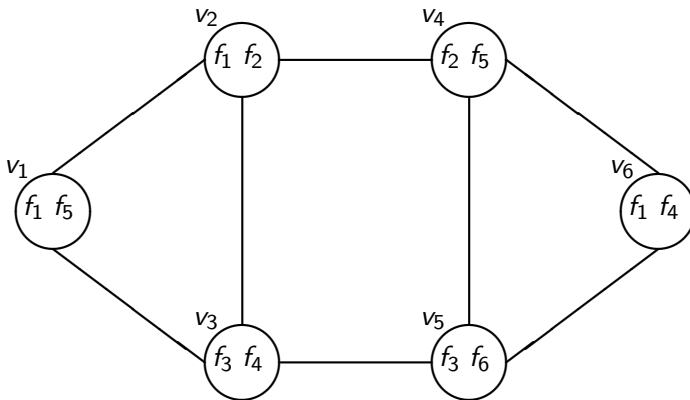
$$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$$

$$K = \{v_1, v_3, v_4\} \subset V$$

## Nachteil der $K$ -Terminal-Reliability

- ▶ Die *K-Terminal-Reliability* (KTR) ist wenig aussagekräftig, da die redundante Verteilung von Daten und Programmen nicht berücksichtigt wird.
- ▶ Besser:
  - ▶ Abstraktion von einer vorgegebene Knotenmenge  $K$ .
  - ▶ Spezifikation einer Menge von Dateien, welche für die erfolgreiche Verarbeitung eines Programms erforderlich sind.

## Beispiel für ein verteiltes System (DPR)



$$F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$$

$$H = \{f_1, f_2, f_6\} \subset F$$



## Notationen

$D = (V, E, F)$	Ungerichteter DCS-Graph mit Knotenmenge $V$ , Kantenmenge $E$ und der Menge $F$ an Dateien, welche in $D$ verteilt sind.
$V$	Menge der Knoten (perfekte Zuverlässigkeit)
$E$	Menge der Kanten, welche statistisch unabhängig voneinander mit bekannter Wahrscheinlichkeit ausfallen können.
$F$	Dateien, welche in $D$ verteilt vorhanden sind. Hierbei erfolgt keine Unterscheidung zwischen Programmen und Daten.
$H \subseteq F$	Definierte Menge von Dateien, welche miteinander kommunizieren können müssen.
$FA_i \subseteq F$	Dateien, welche am Knoten $i$ vorhanden sind.
$p_i$	Zuverlässigkeit von Kante $i$ .
$R(D_H)$	DPR von $D$ bei gegebenem $H$ .

# NP-vollständige Probleme

- ▶  $\mathcal{NP}$  ist die Klasse der Probleme, welche von einer nichtdeterministischen Turingmaschine in Polynomialzeit gelöst werden können.
- ▶ NP-vollständige Probleme sind die *schwersten* Probleme aus  $\mathcal{NP}$ . Um zu zeigen, dass ein Problem  $\pi$  NP-vollständig ist, geht man wie folgt vor:
  1. Zeige, dass  $\pi$  in  $\mathcal{NP}$  liegt.
  2. Wähle ein bekanntes NP-vollständiges Problem  $\pi'$  aus.
  3. Zeige, dass  $\pi'$  mittels einer Transformation  $f$  in Polynomialzeit auf  $\pi$  reduzierbar ist.

# NP-harte Probleme

Ein Problem  $\pi$  heißt NP-hart, falls wir zeigen können, dass  $\pi$  NP-vollständig ist (d. h. wir können eine Transformation  $f$  angeben), nicht aber, dass  $\pi$  in  $\mathcal{NP}$  liegt.

- ▶ Unter der (starken, aber nicht bewiesenen) Annahme, dass  $\mathcal{P} \neq \mathcal{NP}$  gilt, sind NP-harte Probleme nicht effizient, d. h. in Polynomialzeit, lösbar.

# Allgemeine DCS

Die Berechnung der *Distributed Program Reliability* für ein allgemeines Distributed Computing System (DCS) ist NP-hart.

# Planare DCS

Die Berechnung der *Distributed Program Reliability* für ein planares Distributed Computing System (DCS) ist NP-hart.

## DCS mit Sterntopologie

Die Berechnung der *Distributed Program Reliability* für ein Distributed Computing System (DCS) mit Sterntopologie ist NP-hart. Dies gilt auch, falls  $|FA_i| = 2$ , d. h. falls jeder Knoten lediglich zwei Dateien bereithält.

## DCS mit Sterntopologie

Die Berechnung der *Distributed Program Reliability* für ein Distributed Computing System (DCS) mit Sterntopologie ist NP-hart. Dies gilt sogar, wenn es nur zwei Kopien von jeder Datei gibt.

## Weitere Topologien

Die Berechnung der *Distributed Program Reliability* für ein Distributed Computing System (DCS) mit Baumtopologie ist NP-hart.

Die Berechnung der *Distributed Program Reliability* für ein Distributed Computing System (DCS) mit 2-Baum-Topologie ist NP-hart.

Die Berechnung der *Distributed Program Reliability* für ein serien-paralleles Distributed Computing System (DCS) ist NP-hart.



# Fazit

- ▶ Die Zuverlässigkeit in einem DCS drückt sich in der Wahrscheinlichkeit aus, mit der ein verteiltes Programm erfolgreich ausgeführt wird.
- ▶ Die Netzwerktopologie hat großen Einfluss auf die Zuverlässigkeit.
- ▶ Die Berechnung der *Distributed Program Reliability* ist im Allgemeinen NP-hart. Sogar bei einer Einschränkung auf Baum-, Stern-, 2-Baum- oder serien-parallele Topologien bleibt die Lösung dieses Zuverlässigkeitsproblems NP-hart.
- ▶ Es ist somit nicht zu erwarten, dass es eine exakte und effiziente Lösung dieses Problems gibt.

# Fragen?

Vielen Dank für Ihre Aufmerksamkeit.