Dependable Systems (VU 182.712)

Praktisches Ubungsbeispiel SS2015: "Zuverlaessigkeitsmodellierung mit *sharpe*"

Datum der Laboruebung: 18.06.2015

Matr. Nr.	**Name**
1228774	Schieber Constantin
1325582	Pacheiner Peter
1226314	Hofer David

1 Abstract

Ein Computernetzwerk soll durch Markovketten modelliert werden um es auf Fehlersicherheit zu überprüfen.
Konkret soll die Mean Time to Failure (MTTF) und die Verfügbarkeit zwischen 2 Systemen evaluiert und verglichen werden. Eines ohne Redundanz und eines mit. Zusätzlich soll ein Vergleich zwischen den Kosten fuer die beiden Systeme durchgeführt werden.

2 Aufgabenstellung

Für beide Systeme gelten die folgenden Fehlerraten (λ) und Reparaturraten (μ).

- $\lambda_R = 10^{-4}/Std.$ fuer Rechner
- $\lambda_N = 2 * 10^{-5}/Std.$ fuer Switches
- $\mu = 10^{-2}/Std.$
  Um funktionsfaehig zu sein benoetigt das Netzwerk mindestens einen Switch und drei Rechner.

3 Mean Time to Failure

3.1 Einfaches System

Das Computernetzwerk besteht aus drei über einen Switch verbunden Rechnern.
Um die MTTF zu evaluieren reicht es das System durch 2 Zustaende zu modellieren.
Einer stellt den funktionierenden Zustand (0) dar, der andere den Fehlerzustand (1). Die Wahrscheinlichkeit fuer einen Uebergang in den Fehlerzustand setzt sich dann aus $3*\lambda_R+\lambda_N$ zusammen. Diese Vereinfachung kann vorgenommen werden da jeder Ausfall, egal welcher Art, in einen Fehlerzustand fuehrt. Die Reparaturrate muss nicht berücksichtigt werden da wir uns im Moment ausschließlich fuer die MTTF interessieren.
Das System kann sowohl graphisch als auch in Textform beschrieben werden:

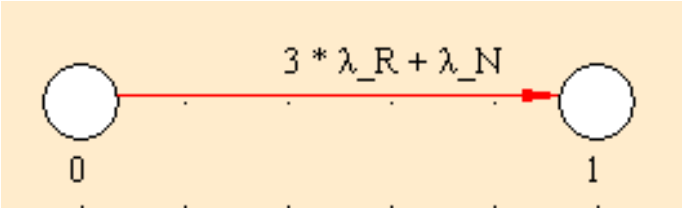

Abbildung 1: Einfaches System in *sharpe* modelliert

```
*Description of the simple model
markov simple
0 1 lambda_N + 3*lambda_R
end
```

3.2 Redundantes System

Das redundante Computernetzwerk besteht aus 4 Rechnern und 2 Switches was die Ausfallwahrscheinlichkeit gegenueber dem einfachen System deutlich verringern sollte.
Statt 2 Knoten werden nun 5 Knoten benoetigt um alle Uebergaenge korrekt abzubilden.
Das System ist in den Knoten 0,1,2,3 funktionsfaehig, Knoten 4 ist der Fehlerzustand (siehe Grafik 2).

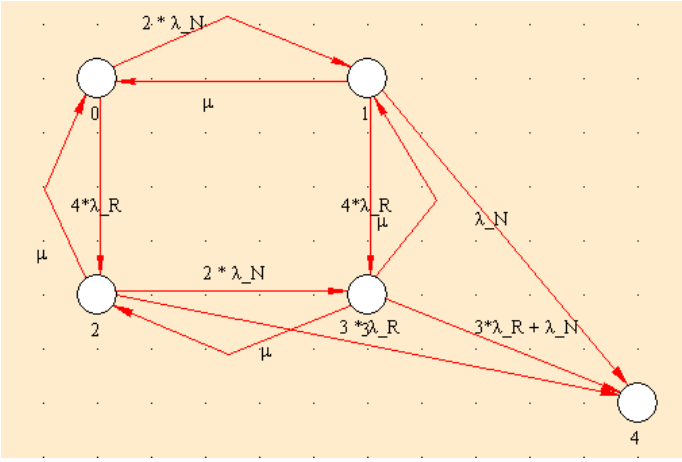

Abbildung 2: Redundantes System in *sharpe* modelliert

```
*Description of the redundant model
markov redundant

2 0 mu
3 2 mu
1 3 mu
1 0 mu
0 1 2*lambda_N
0 2 4*lambda_R
2 3 2*lambda_N
1 3 4*lambda_R
3 4 3*lambda_R + lambda_N
1 4 lambda_N
2 4 3*lambda_R
end
```

3.3 Vergleich der beiden Systeme

Um die beiden Systeme vergleichen zu koennen muss nun auch die Simulation derselbigen durchgeführt werden. Dies geschieht durch folgende Befehlssequenz in *sharpe*:

```
bind
lambda_N      2/100000
lambda_R      1/10000
mu            1/100
end
*Description of the simple model
markov simple
0 1 lambda_N + 3*lambda_R
end
*Starting probability
0 1.0
end
*Description of the redundant model
markov redundant
2 0 mu
3 2 mu
1 3 mu
1 0 mu
0 1 2*lambda_N
0 2 4*lambda_R
2 3 2*lambda_N
1 3 4*lambda_R
3 4 3*lambda_R + lambda_N
1 4 lambda_N
2 4 3*lambda_R
end
*starting probability
0 1.0
end
* calculate MTTF
expr mean(simple)
expr mean(redundant)
```

Daraus resultieren die folgenden Ergebnisse:

```
---------------------------------------------------
mean(SIM):     3.1250e+003
---------------------------------------------------
mean(RED):     8.8540e+004
---------------------------------------------------
```

Das bedeutet dass das einfache Modell nach ca. 31250 Stunden ausfällt und das redundante erst nach 88540 Stunden. Man sieht dass sich durch das hinzufügen der Redundanz der Zeitraum bis zum Ausfall deutlich erhoeht.

4 Availability

Im Gegensatz zur MTTF ist bei der Availability die Reparaturrate nun auch vom Fehlzustand aus zu berücksichtigen.

4.1 Einfaches System

Für das einfache System wird einfach eine neue Kante die zurueck in den funktionierenden Zustand fuehrt hinzugefuegt.

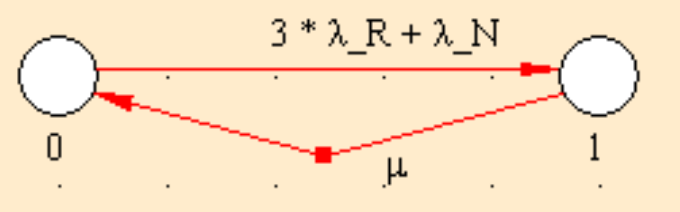

Abbildung 3: Einfaches System mit neuer Kante in *sharpe* modelliert

```
*Description of the simple model with new repair edge
markov simple
0 1 lambda_N + 3*lambda_R
1 0 mu
end
```

4.2 Redundantes System

Im redundanten System reicht nun ein Fehlerzustand nicht mehr aus da wir aus diesem auch wieder zurueck kehren koennen. Um sich den Zustand des restlichen Systems zu merken"bzw. um deterministisch agieren zu koennen werden 3 neue Knoten eingefuehrt, die jeweils als Fehlzustaende fungieren und ueber eine Kante mit dem letzten funktionierenden Zustand verbunden sind.

```
*Description of the redundant model with new failstates
*and edges for repairrates
markov redundant
2 0 mu
3 2 mu
1 3 mu
1 0 mu

0 1 2*lambda_N
0 2 4*lambda_R
2 3 2*lambda_N
1 3 4*lambda_R

*Create new failstates for a deterministic approach
3 34 3*lambda_R + lambda_N
1 14 lambda_N
```

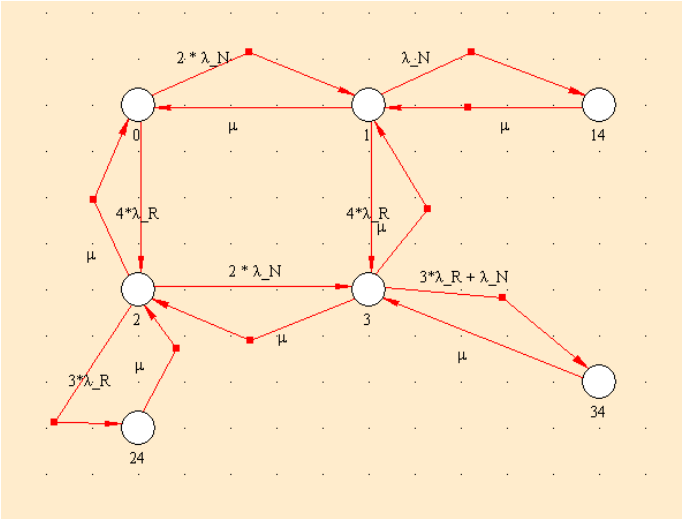

Abbildung 4: Redundantes System mit neuen Knoten und Kanten in *sharpe* modelliert

```
2 24 3*lambda_R

*Add new edges for the repairrates
34 3 mu
14 1 mu
24 2 mu
end
```

4.3 Vergleich der beiden Systeme

Der Vergleich der beiden Systeme wird mit dem Befehl *prob(SYS, NODE)* ausgeführt. Dieser errechnet die Zustandswahrscheinlichkeit fuer eine bestimmte *NODE* im angegebenen System.
Im einfachen System reicht es die Wahrscheinlichkeit berechnen zu lassen mit der sich das System im gültigen Zustand befindet.
Im redundanten Fall ist es einfacher die Wahrscheinlichkeit der Fehlzustände berechnen zu lassen, diese zu addieren und von 100% abzuziehen.

Der Aufbau der Befehlsabfolge sieht so aus:

```
bind
lambda_N      2/100000
lambda_R      1/10000
mu            1/100
end
*Description of the simple model
markov simple
0 1 lambda_N + 3*lambda_R
1 0 mu
end
*Starting probability
0 1.0
end
*Description of the redundant model
markov redundant
2 0 mu
3 2 mu
1 3 mu
1 0 mu
0 1 2*lambda_N
0 2 4*lambda_R
2 3 2*lambda_N
1 3 4*lambda_R
3 34 3*lambda_R + lambda_N
1 14 lambda_N
2 24 3*lambda_R
34 3 mu
14 1 mu
24 2 mu
end
*starting probability
0 1.0
end
*calculate Availability
expr prob(simple, 0)
expr 1-(prob(redundant, 24) + prob(redundant, 34) + prob(redundant, 14))
end
```

Und das Ergebnis dann folgendermaßen:

```
prob(simple, 0):
    9.6899e-001
```

```
1-(prob(redundant, 24) + prob(redundant, 34) + prob(redundant, 14)):
    9.9872e-001
```

Wie man sieht beträgt die Wahrscheinlichkeit dafür dass sich das einfache System in einem gültigen Zustand befindet 96.899%.
Die Wahrscheinlichkeit in einem redundanten System beträgt 99.87%.

5 Kosten