

sAnalyze-Collection¹⁾

Die folgende Dokumentation zum Programmpaket sAnalyze-Collection gliedert sich in die Abschnitte:

- 1 Einleitung**
- 2 Kurzbeschreibung von *sAnalyze-Collection***
- 3 Skriptsprache von *sAnalyze-Collection***
- 4 Anwendungsbeispiel**
- 5 Ergebnisse**
- 6 Zusammenfassung**

Literatur

1) © *sAnalyze-Collection* ist ein an der Fachhochschule Wiesbaden entwickeltes Simulationswerkzeug zur Analyse des Zuverlässigkeitsverhaltens dynamischer Systeme. Neben der Ermittlung des Verfügbarkeitsverlaufes ermöglicht *sAnalyze-Collection* auch die Bestimmung von Aufenthaltswahrscheinlichkeiten und Lebensdauern. Dabei basiert der Lösungsalgorithmus auf einer transienten Zeitanalyse im Zustandsraum, was zur Folge hat, dass sowohl die stationären Verhältnisse auch das Einschwingverhalten durch geeignete Gütemaße zum Ausdruck gebracht werden können.

1 Einleitung

Sowohl zuverlässigkeits- als auch unternehmenskritische Anwendungen (z. B. Verkehrsüberwachung, Versorgungsunternehmen, etc.) erfordern spezielle Maßnahmen, um die erforderliche Verlässlichkeit sicherstellen zu können. Insbesondere muss die ununterbrochene Funktionstüchtigkeit während des Einsatzes, d. h. eine extrem hohe Zuverlässigkeit, gewährleistet werden. Hierbei kommt es weniger auf die stationäre Sicherheitsgüte im eingeschwungenen Zustand als vielmehr auf das transiente (zeitabhängige) Zuverlässigkeits- und Sicherheitsverhalten der eingesetzten Systeme an.

Um die Zuverlässigkeit eines verteilten Systems anhand konkreter Gütemaße effizient beurteilen zu können, wurde in den letzten Jahren mit *sAnalyze-Collection* ein Programmwerkzeug zur Zuverlässigkeits- bzw. Verfügbarkeits- und Lebensdauerbestimmung entwickelt (Prototyp-Entwicklung). Im Rahmen eines weiteren Projektes werden die Aktivitäten unter dem Aspekt *Messverfahren für IT-Sicherheit* kontinuierlich weitergeführt.

Es gibt grundsätzlich verschiedene Möglichkeiten, die Leistung und Zuverlässigkeit einer Netzwerkinfrastruktur abzuschätzen. Für die in der Planungs- und Entwurfsphase häufig anzutreffende Analyse oder Simulation wird ein Modell des Bewertungsobjektes benötigt. Die Modellbildung erfolgt in der Praxis dadurch, dass zunächst ein konzeptionelles Architekturmodell entworfen wird. Aus diesem wird ein formales Modell gewonnen, das dann in ein von einem Rechner analysierbares Modell übertragen bzw. in einer geeigneten Simulationssprache implementiert wird. Bei den heute üblichen Implementierungen wird fast ausnahmslos die Stationarität des Modells vorausgesetzt. Dies führt bisweilen zu unbefriedigenden Abschätzungen und Leistungsangaben. Auch in den durchgeführten Drittmittelprojekten zeigte sich bei den Zuverlässigkeitsbetrachtungen, dass es weniger auf den eingeschwungenen Zustand als vielmehr auf das transiente (zeitabhängige) Verhalten des Systems ankommt. Hieraus stellt sich die Aufgabe, ein Simulationsprogramm zu entwickeln, welches auch außerhalb von statischen Modellannahmen und Ersatzschaltbildern zu zufriedenstellenden Ergebnissen, d. h. zu einer guten Übereinstimmung von Vorhersage und anschließender Beobachtung im realen Umfeld führt.

Mit *sAnalyze-Collection* wurde eine allgemein gültige Methodik zur Analyse des dynamischen Zustandsverhaltens der in Wechselwirkung stehenden Einzelkomponenten umgesetzt. Neben den Methoden der Modellbildung und der Herleitung der algorithmischen Berechnungsverfahren wurden auch die Entwurfskonzepte für eine rechnergestützte Sys-

temanalyse integriert. Aufbauend auf diesen Konzepten wurde dann eine Softwareanwendung erstellt, welche anhand der Modellbildung und der algorithmischen Berechnungsverfahren eine Vorhersage des Zustandsverhaltens sowie die Berechnung der gewünschten Zuverlässigkeitskenngrößen (z. B. der Verfügbarkeit oder der Lebensdauer) während der Laufzeit des betreffenden Systems erlaubt.

2 Kurzbeschreibung von *sAnalyze-Collection*

Das Analysetool besteht aus den beiden Hauptkomponenten Berechnungseinheit und Visualisierungseinheit (Hybrid-Programm). Die gestartete Visualisierungseinheit ruft zunächst die Berechnungseinheit auf und führt den hierin implementierten Berechnungsalgorithmus aus. Die Berechnungseinheit liest die Systembeschreibungdatei, berechnet die für das spezifizierte System ausgewiesenen Zuverlässigkeitsgrößen und speichert die berechneten Daten in den Berechnungsdateien. Nun liest die Visualisierungseinheit die berechneten Daten ein und stellt diese in Tabellenform und in graphischer Form dar. Danach kann das System vom Anwender seinen Anforderungen entsprechend analysiert werden. Zur Analyse stehen Funktionen wie Ein- und Ausblenden von Funktionskurven, Vergrößern und Verkleinern der graphischen Darstellung und selektieren von Teilbereichen zur speziellen Betrachtung der berechneten Werte. Ist ein System analysiert, kann ein Report des aktuellen Zustandes in eine HTML-Datei generiert werden. Abbildung 1 zeigt das entsprechende Programmablaufmodell.

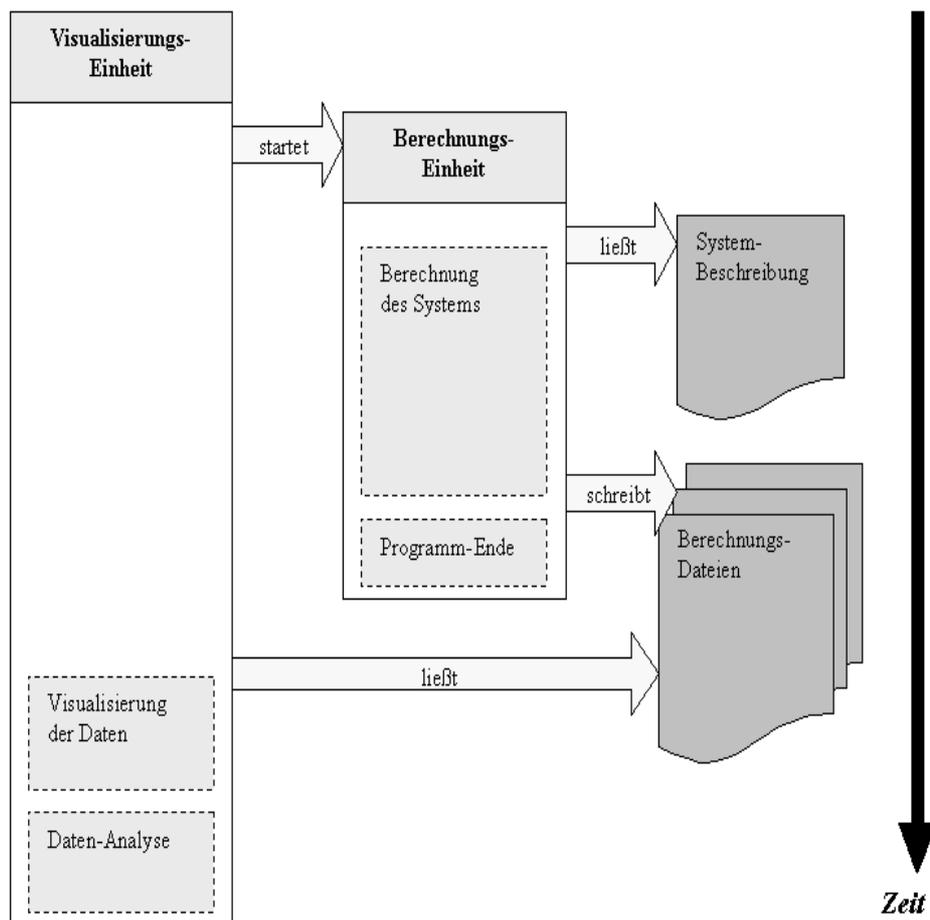


Abbildung 1: Programmablaufmodell

3 Skriptsprache von *sAnalyze-Collection*

Die Systembeschreibungsdatei eines zu analysierenden Systems charakterisiert das Systemmodell und enthält alle für die numerische Berechnung des Modells benötigten Eingabewerte. Um die Systembeschreibungsdatei eindeutig zu beschreiben, wurden die in **Tabelle 1** zusammengestellten syntaktischen Konventionen festgelegt. Mit dem Element „!*text*!“ können an beliebiger Stelle innerhalb der Systembeschreibungsdatei Kommentare zur weiteren Erklärung angegeben werden. Die folgenden Elemente sind in der hier beschriebenen Reihenfolge zu verwenden. Als erste Angabe wird die Knotenzahl definiert. Die Knotenzahl wird mit dem Element „K *zahl*“ beschrieben. Als nächste Angabe werden die Ausfallraten mit dem Element „F *i* *j* *wert*“ definiert. Hierbei bezeichnen *i* und *j* zwei Zustandsknoten, während *wert* die Ausfallrate zwischen genau diesen beiden Zuständen spezifiziert. Dementsprechend werden die Reparaturraten mit dem Element „R *i* *j* *wert*“ definiert. Hierbei gelten für die Parameter die gleichen Wertebereiche wie für die Definition der Ausfallraten. Die Definition der System-Verfügbarkeit wird mit dem Element „V [0|1] {0|1}“ bewerkstelligt, wobei eine 1 in der Parameterliste einen intakten System-Zustand und eine 0 einen defekten System-Zustand kennzeichnet. Um das Berechnungsintervall festzulegen, stehen die Elemente „TS *wert*“ für die Definition des Startzeitpunktes und „TE *wert*“ für die Definition der Endzeitpunktes zur Verfügung. Die Parameter *wert* sind in beiden Fällen beliebige Double-Werte. Die Integrationsschrittweite gibt den Zeitabstand zweier aufeinanderfolgender Näherungsberechnungen innerhalb des zuvor festgelegten Berechnungsintervalls an. Definiert wird die Anfangsschrittweite mit dem Element „H *wert*“. Mit den beiden Elementen „HI *wert*“ und „HX *wert*“ werden der Minimalwert bzw. der Maximalwert der Integrationsschrittweite festgelegt. *wert* ist hier jeweils ein Double-Wert. Dann folgt die Vorgabe bezüglich des einzuhaltenden Diskretisierungsfehlers. Die Elemente „EA *wert*“ und „ER *wert*“ spezifizieren den absoluten bzw. den relativen Diskretisierungsfehler, wobei *wert* jeweils ein beliebiger Double-Wert darstellt. Diese beiden Parameter sind verantwortlich für die Berechnung der Integrationsordnung und der Integrationsschrittweite, welche sich in jedem Zeitschritt ändern können und optimal an die Vorgaben des Diskretisierungsfehlers angepasst werden. Um den Anfangszustand definieren zu können, dient das Element „P*i* *wert*“. Es definiert den zum Startzeitpunkt gültigen Wert der Zustandswahrscheinlichkeit des Zustandsknoten *i*. Der Parameter „*i*“ ist somit ein Integer-Wert und der Parameter „*wert*“ ist ein Double-Wert. Um die Berechnungsart flexibel steuern zu können, werden noch die Elemente „Availability“, „Unavailability“, „Lifetime“, „Filling“ und „All“ eingeführt. Mit dem Element „Availability“ wird die Berechnungseinheit

angewiesen, ausschließlich die System-Verfügbarkeit zu berechnen. Entsprechend wird mit dem Element „Unavailability“ die Ermittlung der Unverfügbarkeit und mit dem Element „Lifetime“ die Ermittlung der mittleren Lebensdauer des Systems mitgeteilt. „Filling“ dient der Berechnung einer fiktiven mittleren Speicherlänge (Warteschlangenmodell). Ist die Berechnung sämtlicher Zuverlässigkeitsgrößen erwünscht, bedient man sich dem Element „All“. Tabelle 1 zeigt alle Elemente nochmals im Überblick.

Element	Bedeutung	Beispiel
! <text> !	Kommentar	! Exercise 1.6 !
K <zahl>	Anzahl der Zustände [$0 \dots zahl$])	K 5
F <i> <j> <wert>	Ausfallraten zw. Zustand i und j	F 0 1 0.0001
R <i> <j> <wert>	Reparaturraten zw. Zustand i und j	R 5 7 0.0025
V [0/1] { 0/1}	Definition der Verfügbarkeit	V 1 0 1 0 0 0
TS <wert>	Startzeitpunkt in ZE	TS 0.0
TE <wert>	Endzeitpunkt in ZE	TE 10.0
H <wert>	Anfangsschrittweite in ZE	H 0.01
HI <wert>	Minimale Schrittweite in ZE	HI 0.001
HX <wert>	Maximale Schrittweite in ZE	HX 0.1
EA <wert>	Absoluter Diskretisierungsfehler	EA 0.01
ER <wert>	Relativer Diskretisierungsfehler	ER 0.1
P<i> <wert>	Anfangswerte ($\sum \forall P_i = 1$)	P4 0.24
Availability Unavailability Lifetime Filling All	Berechnung der Verfügbarkeit, Berechnung der Unverfügbarkeit, Berechnung der Lebensdauer, Berechnung der Speicherlänge oder Berechnung aller Daten	Availability

Tabelle 1: Syntax der Konfigurations-Datei

4 Anwendungsbeispiel

Als Beispiel für die Realisierung einer verteilten Kommunikationsinfrastruktur zeigt Abbildung 2 die Ringstruktur eines ATM/SDH-Backbone-netzes. Neben der Fehlertoleranz der Einzelringe besteht der Backbone aus vier identischen Vermittlungsknoten, die das Verkehrsaufkommen paarweise an zwei entfernt liegenden Standorten (hier Ort A und B) ebenfalls fehlertolerant bewältigen müssen. Die Untersuchung erfolgt unter der Voraussetzung, dass sich die einzelnen SDH-Ringe nicht gegenseitig beeinträchtigen und die erforderlichen Reparaturen möglich sind. Für das eingesetzte technische Equipment gelte folgende Spezifikation (Tabelle 2):

Komponente	Ausfallrate	Reparaturrate
Netzabschluss	$4 \cdot 10^{-3} \text{ h}^{-1}$	1 h^{-1}
Vermittlungsknoten	$8 \cdot 10^{-3} \text{ h}^{-1}$	0.2 h^{-1}

Tabelle 2: Spezifikation des Backbone-netzes

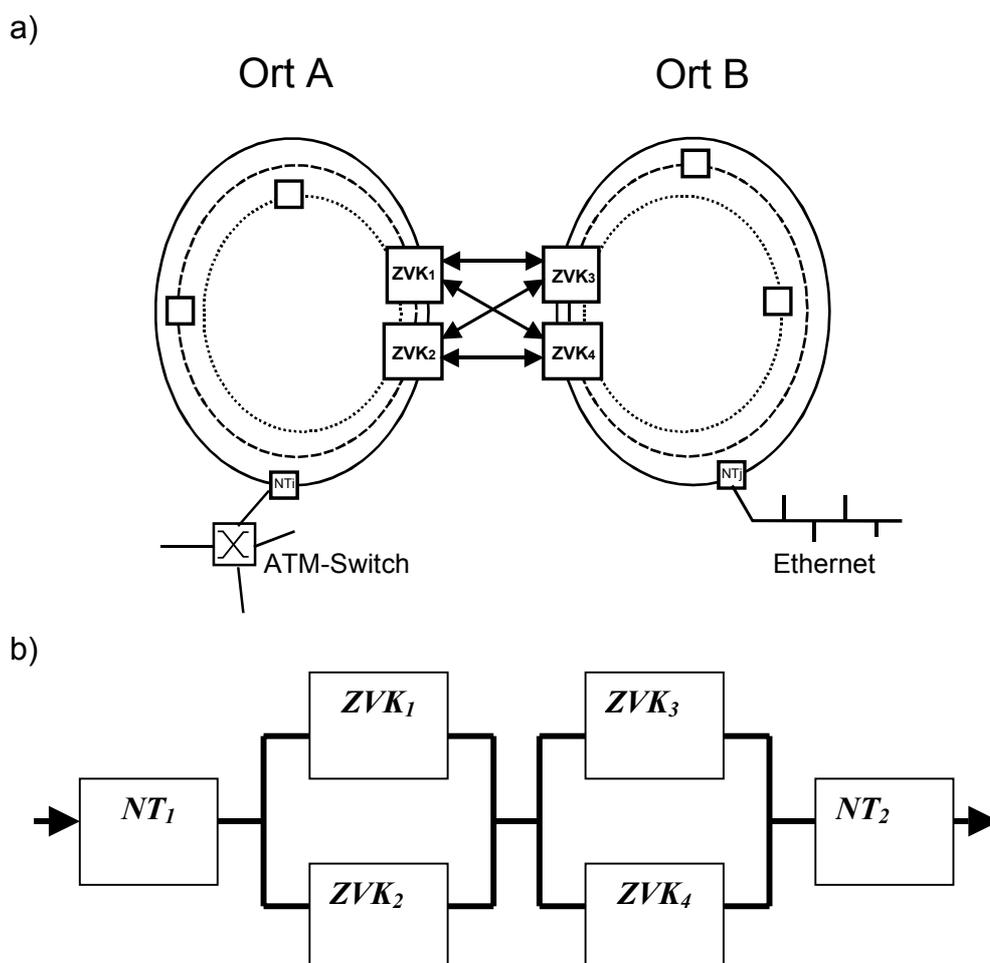


Abbildung 2: Struktur des Backbone-netzes
a) Wirkschaltplan b) Zuverlässigkeits-Blockschaltplan

Das sich aus der Fehlertoleranz der gesamten Netzinfrastruktur ergebende Zustandsmodell ist in Abbildung 3 skizziert. Es existieren insgesamt achtzehn Systemzustände. Die Werte für die Modellparameter (Ausfall- bzw. Reparaturraten) betragen:

$$\lambda_{NT} = 4 \cdot 10^{-3} h^{-1}, \mu_{NT} = 1 h^{-1}, \lambda_{ZK} = 8 \cdot 10^{-3} h^{-1} \text{ und } \mu_{ZK} = 0.2 h^{-1}.$$

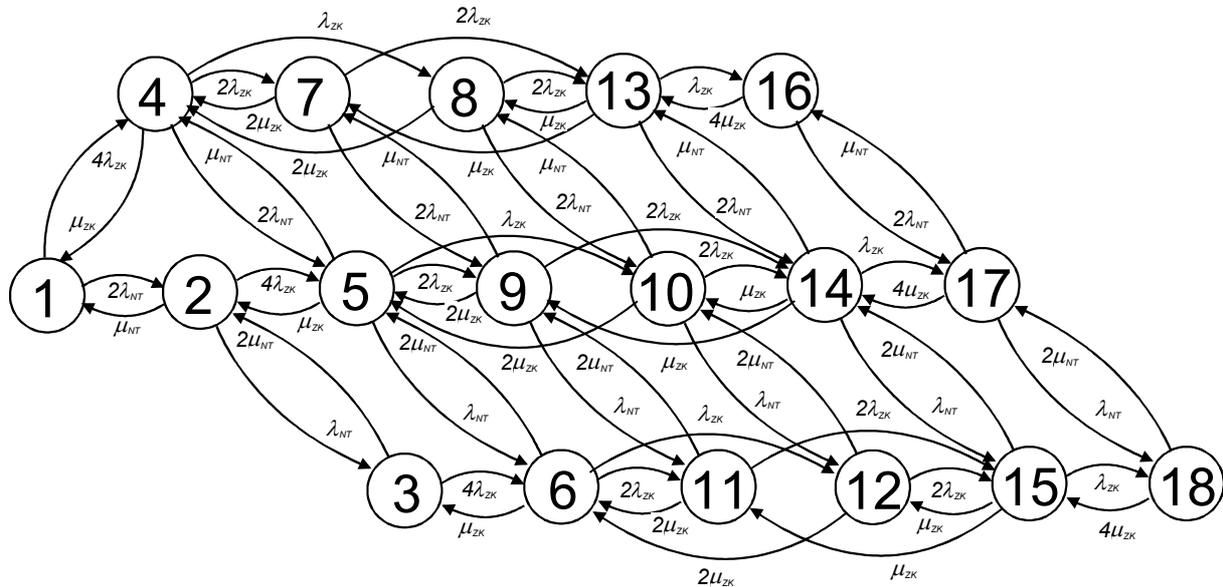


Abbildung 3: Zustandsmodell des Rechnernetzes

Die Anfangs-Wahrscheinlichkeiten zum Zeitpunkt $t = 0$ werden wie folgt angenommen:

$$P_1(0) = 0.5, P_2(0) = 0.25, P_3(0) = 0.25 \text{ und } P_4(0) \dots P_{18}(0) = 0.$$

Bei einer gewählten Integrationsschrittweite von $\Delta t = 0.5 h$ ergibt sich für sAnalyse-Collection die nachstehende Eingabebeschreibung (Tabelle 3):

```
! Start of Configuration File !
! Transient State Analysis of a Fault Tolerance System !
! Input File for Exercise IVBB !
! Eingabefile Stand 23.06.2004 fuer Exec mit 18 Knoten !
! Letzter Stand Ergebnis: V(num) = 0.989115 !
! Theor. Wert: V(theo) = 0.989024 !
! File Name: Input.prj !
! Author: Bernhard Geib !
! Date: 20/10/2004 !
! Start of File !
! System States 0, 1 to 17 !
```

! Failure Rates !

F 0 1 0.008
F 0 3 0.032
F 1 2 0.004
F 1 4 0.032
F 2 5 0.032
F 3 4 0.008
F 3 6 0.016
F 3 7 0.008
F 4 5 0.004
F 4 8 0.016
F 4 9 0.008
F 5 10 0.016
F 5 11 0.008
F 6 8 0.008
F 6 12 0.016
F 7 9 0.008
F 7 12 0.016
F 8 10 0.004
F 8 13 0.016
F 9 10 0.004
F 9 11 0.004
F 10 14 0.016
F 11 14 0.016
F 12 13 0.008
F 12 15 0.008
F 13 14 0.004
F 13 16 0.008
F 14 17 0.008
F 15 16 0.008
F 16 17 0.004

! Repair Rates !

R 1 0 1.0
R 2 1 2.0
R 3 0 0.2
R 4 1 0.2
R 4 3 1.0
R 5 2 0.2
R 5 4 2.0
R 6 3 0.4
R 7 3 0.4
R 8 4 0.4
R 8 6 1.0
R 9 4 0.4
R 9 7 1.0
R 10 5 0.4
R 10 8 2.0
R 10 9 1.0
R 11 5 0.4
R 11 9 2.0
R 12 6 0.4

```

R 12 7 0.2
R 13 8 0.4
R 13 9 0.2
R 13 12 1.0
R 14 10 0.4
R 14 11 0.2
R 14 13 2.0
R 15 12 0.8
R 16 13 0.8
R 16 15 1.0
R 17 14 0.8
R 17 16 2.0

! Availability Definition  $V = P0 + P3 + P6$  !
V 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0
! Start and End Time !
TS 0.0
TE 100.0
! Integration Step !
H 0.5
HI 0.1
HX 1.0
! Discretisation Error !
EA 0.0001
ER 0.0001
! Start Values: Sum(Pi) must be 1 !
P0 0.5
P1 0.25
P2 0.25
P3 0.0
P4 0.0
P5 0.0
P6 0.0
P7 0.0
P8 0.0
P9 0.0
P10 0.0
P11 0.0
P12 0.0
P13 0.0
P14 0.0
P15 0.0
P16 0.0
P17 0.0

! Output Definition !
All
! End of File !

```

Tabelle 3: Eingabebeschreibung für sAnalyze-Collection

5 Ergebnisse

Für den stationären Wert der Verfügbarkeit $V_{sta} := V(t \rightarrow \infty)$ erhält man aus dem Zuverlässigkeits-Blockschaltplan (Abbildung 2b) die Beziehung:

$$V_{sta} = (V_{NT})^2 * (V_{ZK} * (2 - V_{ZK}))^2 \quad (22)$$

mit

$$V_{NT} = \frac{\mu_{NT}}{\mu_{NT} + \lambda_{NT}} \quad (23)$$

und

$$V_{ZK} = \frac{\mu_{ZK}}{\mu_{ZK} + \lambda_{ZK}} \quad (24)$$

Dies ergibt:

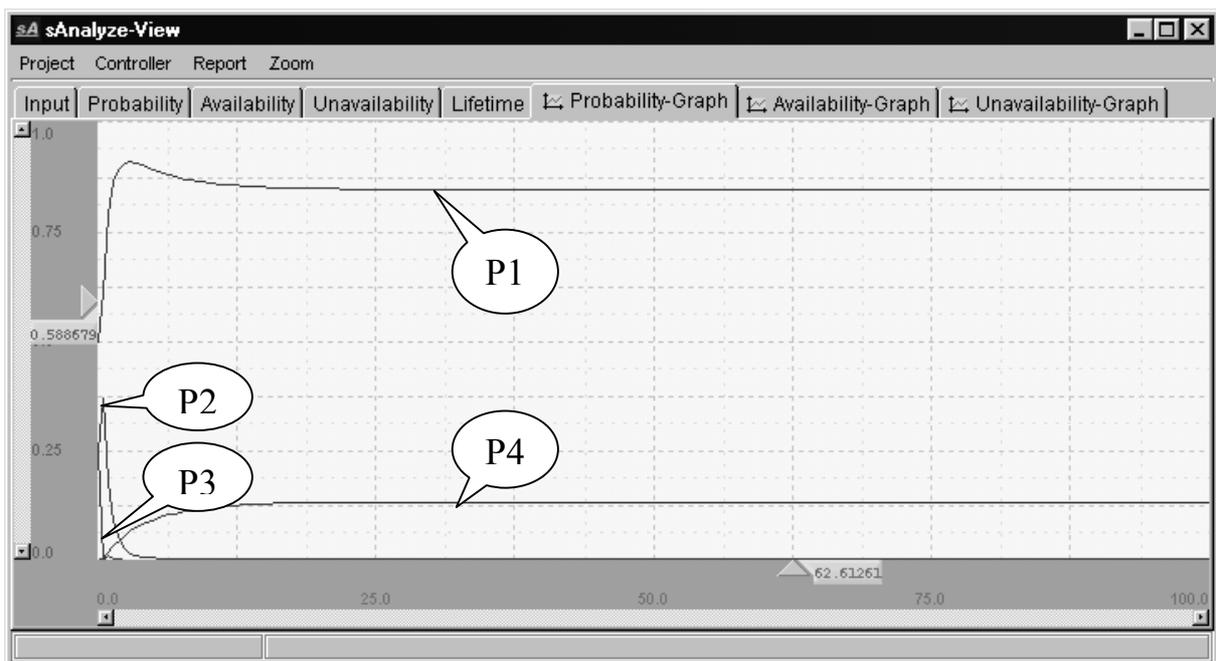
$$\underline{V_{sta} = 0.989024}$$

Der mit dem Analyse-Tool berechnete Werte zum Zeitpunkt $t = 100 h$ ergibt sich zu:

$$\underline{V_{sim} = 0.989115.}$$

Abbildung 4 zeigt den zeitlichen Verlauf der Zustandswahrscheinlichkeiten $P_1(t)$ bis $P_4(t)$ sowie der Systemverfügbarkeit $V(t)$.

a)



b)

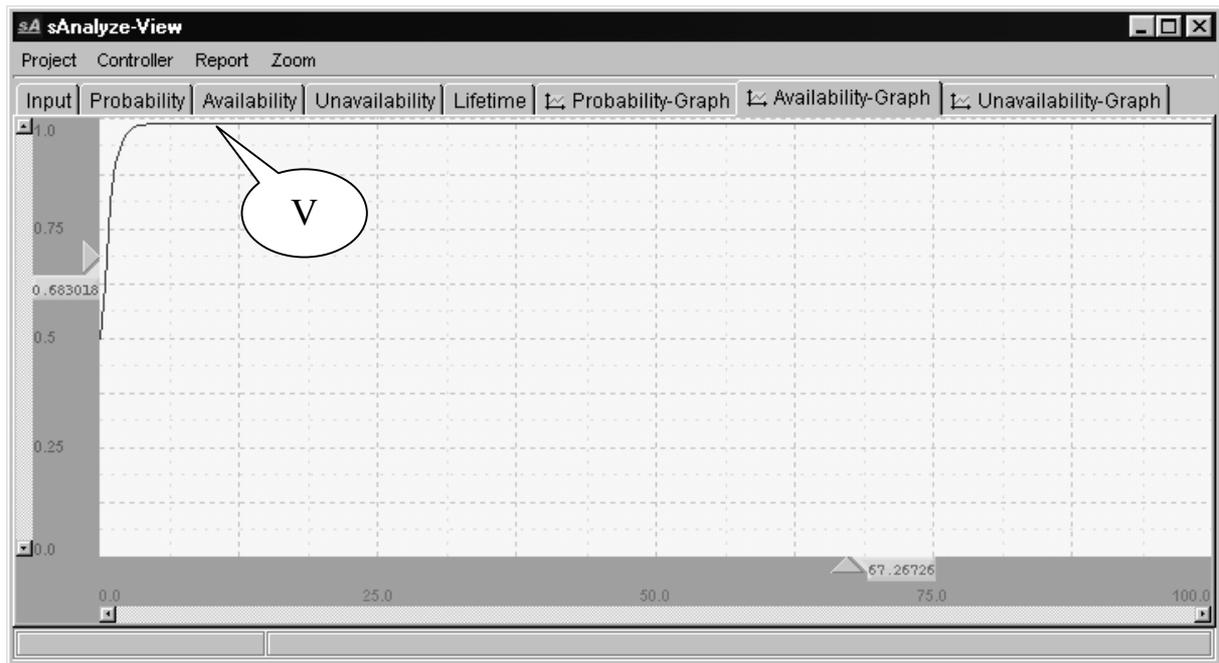


Abbildung 4: a) Wahrscheinlichkeits- und b) Verfügbarkeitsverlauf des Backbonenetzes

Der Vergleich zeigt eine gute Übereinstimmung zwischen der statischen Näherungslösung und der mittels dem Analysetool durchgeführten dynamischen Berechnung.

6 Zusammenfassung

Besonders in Zeiten hohen Kostendrucks gilt es sowohl im privaten als auch im öffentlichen Telekommunikationsbereich, die Network Down Time zu minimieren und gleichzeitig die bestehenden Anforderungen bezüglich der Verfügbarkeit und Ausfallsicherheit einzuhalten. Für den Betreiber eines verteilten Kommunikationssystems bedeutet dies, dass der Ausfall einer Netzwerkkomponente nicht zum Ausfall eines Teilnetzwerkes oder gar des gesamten Kommunikationsverbundes führen darf. In sicherheitskritischen Systemumgebungen wird dies meistens dadurch gelöst, dass die zentralen Netzwerkkomponenten und deren Verbindungen untereinander hochverfügbar bzw. redundant ausgelegt sind. Da der Aufwand für die Projektierung von redundanten Netzwerken dem von einfach ausgelegten Netzwerken entsprechen sollte, sind flexible Analysewerkzeuge und Simulationseinrichtungen unerlässlich, welche die im Betrieb einzuhaltenden Zuverlässigkeitsanforderungen von verteilten Telekommunikationssystemen bereits während der Planungsphase verifizierbar gestalten. Vor diesem Hintergrund war nun Ziel des Beitrags, die Grundlagenarbeiten und Grunduntersuchungen für ein solches Analyse-Tool zu bewerkstelligen. Hierzu sind die Vorgehensweisen bei der Modellbildung ebenso zu betrachten, wie die Implementierung der algorithmischen Berechnungsverfahren und die Integration der graphischen Darstellungselemente.

Die Ausführungen eines Praxisbeispiels demonstriert nicht nur die Handhabung des Analyse-Tools, sondern liefern auch einen Beweis für die hohe Stabilität des Lösungsalgorithmus sowie für die Korrektheit der erzielten Simulationsergebnisse. Durch Vergleich mit analytischen Berechnungen des stationären Endwertes der Verfügbarkeit wurden auch die Ergebnisse der nötigen Kontrolluntersuchungen illustriert.

Literatur:

- [1] Störmer, H.: Mathematische Theorie der Zuverlässigkeit, R. Oldenbourg Verlag München Wien, 1983.
- [2] Geib, B.: Vorlesungsskript Verlässliche Systeme, FHW, SS 2006.
- [3] Schneeweis, W. G.: Zuverlässigkeitstechnik, Datakontext-Verlag, Köln 1992.