

# Vernetzte Systeme

## Lösungsvorschlag 6

### 1 TCP-Zustandsdiagramm

a) Zum Verbindungsaufbau gehören die Nachrichten 1 (SYN), 2 (ACK, SYN) und 3 (ACK). Durch die SYN-Nachrichten in diesem 3-Way-Handshake werden die initialen Sequenznummern ausgehandelt (synchronisiert). Die initiale Sequenznummer des Clients ist 3459 (1, SYN), die des Servers 8656 (2, ACK, SYN).

b) FIN-Nachrichten lösen den Verbindungsabbau in der gesendeten Richtung aus. Der Sender einer FIN-Nachricht gibt damit an, dass er keine weiteren Daten mehr zu senden hat. Trotzdem können in der anderen Richtung noch Daten empfangen werden.

Zum Verbindungsabbau gehören die Nachrichten 9, 10, 11 und 12. Der Server löst den Verbindungsabbau durch das gesetzte FIN-Bit in Nachricht 9 aus. Dieses FIN hat Sequenznummer 9075. Die Nachrichten 10 und 11 bestätigen das FIN. Mit Nachricht 11 schliesst der Client seinerseits die Verbindung. Sein FIN hat Sequenznummer 4136 und wird mit Nachricht 12 vom Server bestätigt.

c) Tabelle mit Zustandsübergängen

**Server:**

Zustand	Eingabe	Ausgabe	Nachfolgezust.
CLOSED	passive OPEN	—	LISTEN
LISTEN	rcv SYN (1)	send SYN,ACK (2)	SYN RCVD
SYN RCVD	rcv ACK (3) of SYN 3459	—	ESTAB
ESTAB	rcv Data (4)	send ACK (5)	ESTAB
ESTAB	SEND	send data (6)	ESTAB
ESTAB	rcv ACK (7)	—	ESTAB
ESTAB	SEND	send data (8)	ESTAB
ESTAB	CLOSE	send FIN 9075 (9)	FIN WAIT-1
FIN WAIT-1	rcv ACK (10) of FIN 9075	—	FIN WAIT-2
FIN WAIT-2	rcv FIN 4136 (11)	send ACK (12)	TIME WAIT
TIME WAIT	Timeout	—	CLOSED

**Client:**

Zustand	Eingabe	Ausgabe	Nachfolgezust.
CLOSED	active OPEN	send SYN (1)	SYN SENT
SYN SENT	rcv SYN,ACK (2)	send ACK (3)	ESTAB
ESTAB	SEND	send data (4)	ESTAB
ESTAB	rcv ACK (5)	—	ESTAB
ESTAB	rcv data (6)	send ACK (7)	ESTAB
ESTAB	rcv data (8)	—	ESTAB
ESTAB	rcv FIN 9075 (9)	send ACK (10) of FIN 9075	CLOSE WAIT
CLOSE WAIT	CLOSE	send FIN 4136 (11)	LAST-ACK
LAST-ACK	rcv ACK of FIN 4136 (12)	—	CLOSED

d) Diese Folge wird durchlaufen, wenn beide Seiten (nahezu) gleichzeitig geschlossen werden (CLOSE-Kommando).

## 2 Proxy Server und Warteschlangen

- a) Um zu bestimmen, ob das System stabil ist, müssen wir die Auslastung (load, traffic intensity)  $\rho$  kennen. Bekannt ist, wieviele Pakete durchschnittlich ankommen, also die Ankunftsrate (arrival rate)  $\lambda = 99s^{-1}$ , und wie viele Pakete verarbeitet werden können, also der Durchsatz (throughput)  $\mu = 100s^{-1}$ .

Damit lässt sich die Auslastung  $\rho$  berechnen zu:

$$\rho = \frac{\lambda}{\mu} = \frac{99s^{-1}}{100s^{-1}} = 0.99$$

Da  $\rho = 0.99 < 1$  lässt sich sagen, dass das System stabil ist. Es ist jedoch stark belastet.

- b) Durchschnittlich erwartete Anzahl Pakete  $N$  im Gateway:

$$E[N] = \frac{\rho}{1-\rho} = \frac{0.99}{0.01} = 99$$

- c) Durchschnittlich erwartete Zeit  $T$  im Router:

$$E[T] = \frac{E[N]}{\lambda} = \frac{99}{99s^{-1}} = 1s$$

- d) Durchschnittlich erwartete Zeit  $Q$  in Warteschlange:

$$E[Q] = E[T] - E[X] = E[T] - \frac{1}{\mu} = 1s - \frac{1}{100s^{-1}} = 1s - 0.01s = 0.99s$$

- e) Durchschnittlich erwartete Anzahl Pakete  $N_p$  im Router mit Proxy:

Der Durchsatz  $\mu$  bleibt mit Proxy gleich.

Aber die Ankunftsrate ändert sich zu  $\lambda_p = \frac{10}{11} \cdot \lambda = \frac{10}{11} \cdot 99s^{-1} = 90s^{-1}$

Damit wird die Auslastung  $\rho_p = \frac{\lambda_p}{\mu} = \frac{90s^{-1}}{100s^{-1}} = 0.9$

Und  $E[N_p] = \frac{\rho_p}{1-\rho_p} = \frac{0.9}{0.1} = 9$

- f) Durchschnittliche erwartete Wartezeit  $T_p$  mit Proxy:

$$E[T_p] = \frac{E[N_p]}{\lambda_p} = \frac{9}{90s^{-1}} = 0.1s$$

Das heisst die durchschnittliche Wartezeit hat um einen Faktor 10 abgenommen, obwohl nur  $\frac{1}{11}$  weniger Pakete zu verarbeiten sind!

## 3 Markovkette und Wetterprognose

In der Aufgabenstellung ist gegeben, dass sich unser Wettersystem über lange Zeit eingespielt hat. Es befindet sich also in einen stabilen Zustand (steady state), welcher nicht mehr vom Anfangswert abhängig ist. Für diesen Gleichgewichtszustand gilt, dass die Wahrscheinlichkeit, dass sich das System in einem Zustand befindet nur von den Übergangswahrscheinlichkeiten abhängt.

In unserm Modell sei  $p(1)$  die Wahrscheinlichkeit, dass wir uns im Zustand 'Sonne' befinden, und entsprechend  $p(2)=p(\text{'Nebel'})$ ,  $p(3)=p(\text{'Regen'})$  und  $p(4)=p(\text{'Schnee'})$ .

Für  $p(1)$  lässt sich nun folgende Gleichung aufstellen:

$$p(1) \cdot 0.55 + p(2) \cdot 0.3 + p(3) \cdot 0.3 = p(1)$$

Das heisst, die Wahrscheinlichkeit, dass das System im Zustand 'Sonne' ist, ist gleich der Summe der Wahrscheinlichkeit, dass das System in einem Zeitschritt in den Zustand 'Sonne' wechselt. Und wie gross diese Wahrscheinlichkeiten sind, ist im Diagramm auf der Aufgabenstellung gegeben. Das System ist also zum Beispiel mit  $p(2)$  im Zustand 'Nebel' und wechselt mit Wahrscheinlichkeit 0.3 in den Zustand 'Sonne' und mit Wahrscheinlichkeit  $p(1)$  war es im Zustand 'Sonne' bleibt mit Wahrscheinlichkeit 0.55 auch im nächsten Zeitschritt dort.

Analog lassen sich Gleichungen für  $p(2)$ ,  $p(3)$  und  $p(4)$  aufstellen:

$$p(1) \cdot 0.25 + p(2) \cdot 0.30 + p(3) \cdot 0.25 + p(4) \cdot 0.35 = p(2)$$

$$p(1) \cdot 0.20 + p(2) \cdot 0.25 + p(3) \cdot 0.35 + p(4) \cdot 0.15 = p(3)$$

$$p(0) \cdot 0.00 + p(2) \cdot 0.15 + p(3) \cdot 0.10 + p(4) \cdot 0.50 = p(4)$$

Somit erhalten wir vier Gleichungen mit den vier unbekanntem Wahrscheinlichkeiten  $p(1)$  bis  $p(4)$ . Dieses Gleichungssystem lässt sich in Matrixschreibweise wie folgt darstellen:

$$W \cdot p = p$$

was äquivalent ist zu

$$W \cdot p = I \cdot p$$

wobei I die 4x4 Einheitsmatrix ist. Dies lässt sich umformen zu:

$$(W - I) \cdot p = 0$$

Dieses System ist nicht eindeutig lösbar. [Angenommen,  $p$  wäre die einzige Lösung von  $W \cdot p = p$ , dann wäre  $q = \alpha \cdot p$ ,  $\alpha \in \mathbb{R}$  auch eine Lösung. Denn  $W \cdot q = W \cdot \alpha \cdot p = \alpha \cdot W \cdot p \stackrel{!}{=} \alpha \cdot p \Rightarrow W \cdot q = q$ ] Als zusätzliche Bedingung geben wir noch an, dass die Summe der Wahrscheinlichkeiten aller Zustände 1 sein muss. Also in unserem Fall:

$$\sum_{i=1}^4 p(i) \stackrel{!}{=} 1$$

Somit wird unser Gleichungssystem zu:

$$\begin{pmatrix} W & - & I \\ & & \\ & & \\ & & \\ 1 & 1 & 1 & 1 \end{pmatrix} \cdot p = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Diese lässt sich nun beispielsweise mittels Matlab lösen:

```
>> W = [.55 .3 .3 0; .25 .3 .25 .35; .2 .25 .35 .15; 0 .15 .1 .5]
```

```
W =
```

```
    0.5500    0.3000    0.3000         0
    0.2500    0.3000    0.2500    0.3500
    0.2000    0.2500    0.3500    0.1500
         0     0.1500    0.1000    0.5000
```

```
>> Wp = [W - eye(4); [1,1,1,1]]
```

```
Wp =
```

```
   -0.4500    0.3000    0.3000         0
    0.2500   -0.7000    0.2500    0.3500
    0.2000    0.2500   -0.6500    0.1500
         0     0.1500    0.1000   -0.5000
    1.0000    1.0000    1.0000    1.0000
```

```
>> p = Wp \ [0;0;0;0;1]
```

```
p =
```

```
    0.3472
    0.2770
    0.2438
    0.1319
```

Die Wahrscheinlichkeit, dass ein zufälliger Tag ein Nebeltag ist, ist also 0.277. [Leider berücksichtigt unser Modell nicht, dass es im Herbst/Winter in Zürich öfters Nebel hat als zu andern Jahreszeiten! ;-)]

## 4 Birth-Death-Markov-Prozesse

- a) Wir zeigen die Formel per Induktion. Zur Verankerung zeigen wir als erstes, dass die Formel für  $p_0$  und  $p_1$  gilt. Bei  $p_0$  ist der Fall klar; um  $p_1$  zu erhalten, stellen wir die Gleichgewichtsbedingung für den Zustand 0 auf (die Wahrscheinlichkeit in den Zustand 0 zu kommen ist gleich gross wie die Wahrscheinlichkeit, den Zustand 0 zu verlassen):

$$p_1 \cdot \mu_1 t = p_0 \cdot \lambda_1 t \implies p_1 = \frac{\lambda_1}{\mu_1} p_0.$$

Die Gleichgewichtsbedingung für einen Zustand  $i \geq 1$  erlaubt uns nun von  $p_{i-1}$  und  $p_i$  auf  $p_{i+1}$  zu schliessen (Induktionsschritt):

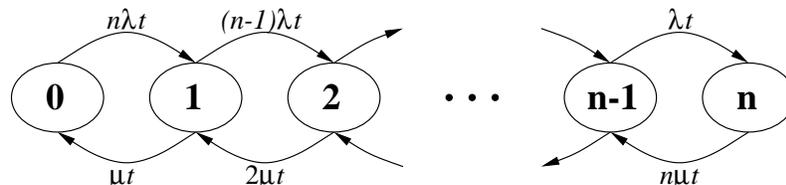
$$p_{i+1} \cdot \mu_{i+1}t + p_{i-1} \cdot \lambda_i t = p_i \cdot (\lambda_{i+1}t + \mu_i t).$$

Wir lösen nach  $p_{i+1}$  auf und setzen für  $p_{i-1}$  und  $p_i$  die Formel (Induktionsannahme) ein:

$$\begin{aligned} p_{i+1} &= \frac{\lambda_{i+1}}{\mu_{i+1}} p_i + \frac{\mu_i}{\mu_{i+1}} p_i - \frac{\lambda_i}{\mu_{i+1}} p_{i-1} \\ &= \frac{\lambda_1 \lambda_2 \cdot \dots \cdot \lambda_{i+1}}{\mu_1 \mu_2 \cdot \dots \cdot \mu_{i+1}} p_0 + \frac{\mu_i \lambda_i}{\mu_{i+1} \mu_i} p_{i-1} - \frac{\lambda_i}{\mu_{i+1}} p_{i-1} \\ &= \frac{\lambda_1 \lambda_2 \cdot \dots \cdot \lambda_{i+1}}{\mu_1 \mu_2 \cdot \dots \cdot \mu_{i+1}} p_0. \end{aligned}$$

□

- b) Wie in der Vorlesung besprochen, können wir das Zeitintervall  $t$  zwischen Zustandsübergängen so klein werden lassen, dass immer höchstens ein Spieler aufs Mal seinen „Gesundheitszustand“ (krank werden oder gesund werden) wechselt. Die Wahrscheinlichkeit, dass mehrere Spieler zusammen krank und/oder gesund werden, wird dann vernachlässigbar. Nehmen wir also an,  $k$  Spieler seien gesund. Die Wahrscheinlichkeit, dass ein Spieler krank wird ist dann  $1 - (1 - \mu t)^k = k\mu t + o(t)$ , für genug kleine  $t$  also  $k\mu t$ . Bei  $k$  gesunden Spielern hat es  $n - k$  kranke Spieler und demzufolge ist die Wahrscheinlichkeit, dass einer dieser kranken Spieler wieder gesund wird  $(n - k)\lambda t$ . Wir erhalten folgenden Birth-Death-Markov-Prozess:



- c) Wir können die Übergangswahrscheinlichkeiten aus Aufgabe b) in die Formel aus Aufgabe a) einsetzen:

$$p_i = \frac{n(n-1) \cdot \dots \cdot (n-i+1) \cdot \lambda^i}{1 \cdot 2 \cdot \dots \cdot i \cdot \mu^i} p_0 = \binom{n}{i} \left(\frac{\lambda}{\mu}\right)^i p_0.$$

Um  $p_0$  zu erhalten, müssen wir jetzt noch die Bedingung einsetzen, dass die Summe aller Wahrscheinlichkeiten 1 ist ( $\rho := \lambda/\mu$ ):

$$p_0 \sum_{i=0}^n \binom{n}{i} \rho^i = p_0 (1 + \rho)^n \stackrel{!}{=} 1 \implies p_0 = \frac{1}{(1 + \rho)^n}.$$

Für  $p_i$  erhalten wir nun also

$$p_i = \frac{\binom{n}{i} \rho^i}{(1 + \rho)^n}.$$

- d) Die Wahrscheinlichkeit  $p$ , dass die Mannschaft spielfähig ist, ist  $p_{11} + p_{12} + \dots + p_n$ , für die gegebenen Werte ( $n = 20$ ,  $\rho = 2.5$ ) ergibt sich (z.B. mit Maple)

$$p = 0.965.$$

Maple-Code:

```
> f := (n, rho) -> sum(binomial(n,i)*rho^i, i=11..n)/(1+rho)^n;
                                n
                                sum binomial(n, i) rho^i
f := (n, rho) -> -----
                                (1 + rho)^n
> evalf(f(20, 2.5));
.9651307886
```