Thomas G. Robertazzi

Li Shi

Networking and Computation: Technology, Modeling and Performance—Solutions Manual
April 27, 2020

Springer

2

Fundamental Stochastic Models

Solution 2.1.

$$P_n(t+\Delta t) = P_n(t)P_{n,n}(\Delta t) + P_{n-1}(t)P_{n-1,n}(\Delta t)$$
$$n = 1, 2, \ldots + P_{n+1}(t)P_{n+1,n}(\Delta t)$$
$$P_0(t+\Delta t) = P_0(t)P_{0,0}(\Delta t) + P_1(t)P_{1,0}(\Delta t)$$

These equations relate the probability of n customers at time $t+\Delta t$ to the probabilities of $n-1$, n, and $n+1$ customers at time t. This is a first order model. Then:

$$\begin{aligned} P_n(t+\Delta t) = {} & P_n(t)(1-\lambda(n)\Delta t)(1-\mu(n)\Delta t) \\ & + P_{n-1}(t)\lambda(n-1)\Delta t \\ & + P_{n+1}(t)\mu(n+1)\Delta t \end{aligned}$$
$$P_0(t+\Delta t) = P_0(t)(1-\lambda(0)\Delta t) + P_1(t)\mu(1)\Delta t$$

Making a difference equation:

$$\begin{aligned} \frac{P_n(t+\Delta t) - P_n(t)}{\Delta t} = {} & -(\lambda(n)+\mu(n))P_n(t) \\ & + \lambda(n-1)P_{n-1}(t) \\ & + \mu(n+1)P_{n+1}(t) \quad n = 1, 2, \ldots \end{aligned}$$
$$\frac{P_0(t+\Delta t) - P_0(t)}{\Delta t} = -\lambda(0)P_0(t) + \mu(1)P_1(t)$$

If $\Delta t \to 0$:

$$\begin{aligned} \frac{dP_n(t)}{dt} = {} & -(\lambda(n)+\mu(n))P_n(t) \\ & + \lambda(n-1)P_{n-1}(t) \\ & + \mu(n+1)P_{n+1}(t) \quad n = 1, 2, \ldots \end{aligned}$$
$$\frac{dP_0(t)}{dt} = -\lambda(0)P_0(t) + \mu(1)P_1(t)$$

Solution 2.2. $\lambda = 400$ calls/s

$$t = \frac{1}{400}\text{s},\ \lambda t = 1$$
$$t = \frac{1}{1000}\text{s},\ \lambda t = 0.4$$

$P_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t} \rightarrow$ Poisson distribution

$\underline{n}$	$\underline{\lambda t = 0.4}$	$\underline{\lambda t = 1.0}$
0	0.67	0.368
1	0.268	0.368
2	0.0536	0.184

Solution 2.3. $\lambda = 400$ calls/s

$$\overline{\#\text{calls}} \Rightarrow \lambda t = 400 \times \frac{1}{400} = 1 \text{ call}$$
$$\Rightarrow \lambda t = 400 \times \frac{1}{4} = 100 \text{ calls}$$
$$\Rightarrow \lambda t = 400 \times 1 = 400 \text{ calls}$$

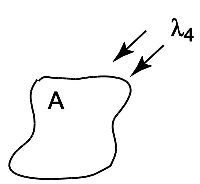
Solution 2.4.

$$\lambda T = 400 \times 0.25 = 100 \text{ calls}$$
$$3\lambda T = 1200 \times 0.25 = 400 \text{ calls}$$

–––––––––––––––––––

300 additional calls

Solution 2.5.



$\lambda_A A = [\#\text{stations/area}][\text{area}] = \#\text{stations}$

Solution 2.6. As i increases, it is less and less likely that the first arrival has not occurred yet.

Solution 2.7. Under a geometric distribution the expected time to the first breakdown is $\frac{1}{P}$ or 500 days. But P may increase with truck age.

Solution 2.8.

P [Train of length i]

$$P[\text{Train of length } i] = P^{i-1}(1-P)$$

$$\overline{\text{Train length}} = \sum_{i=1}^{\infty} iP(i) = \sum_{i=1}^{\infty} i(1-P)P^{i-1}$$

$$= (1-P)\sum_{i=1}^{\infty} iP^{i-1}$$

$$= \frac{(1-P)}{P}\sum_{i=1}^{\infty} iP^{i}$$

$$= \frac{(1-P)}{P}\frac{P}{(1-P)^2} = \frac{1}{1-P}$$

Solution 2.9.

$$\sigma^2 = P(1-P)$$

maximized at $P = \frac{1}{2}$ as the most uncertainty in the random variable at this value.

Solution 2.10.

$$\mathrm{E}[\#\text{ arrivals for 1 slot}] = P$$

As slots are independent in Bernoulli process,

$$E[\#\text{arrivals for N slots}] = N_P$$

Solution 2.11. Use binomial distribution:

$$P[5 \text{ arrivals in } 10 \text{ slots}] = \binom{N}{n} P^n (1-P)^{N-n}$$

$$= \binom{10}{5} P^5 (1-P)^5$$

$$= \binom{10}{5} (0.2)^5 (0.8)^5$$

$$= 252 \times 0.00032 \times 0.32768$$

$$= 0.264$$

Solution 2.12. Because

$$\binom{10}{7} = \binom{10}{3}$$

Solution 2.13.

(a)

$$P = \frac{20}{60} = \frac{1}{3} = 0.33\ q = 1-P$$

(b)

$$
\begin{aligned}
&P(\text{at least one circuit free})\\
&\quad = P(\text{finds a free circuit})\\
&\quad = 1 - P(0 \text{ circuit free})\\
&\quad = 1 - \binom{3}{0} q^{o}(1-q)^{3} = 1 - P^{3}
\end{aligned}
$$

(c)

$$
P(\text{one circuit free}) = \binom{3}{1} q(1-q)^{2}
$$

(d)

$$
\begin{aligned}
\overline{\#\text{busy channels}} &= 1 \cdot P(1\,\text{busy}) + 2P(2\text{ busy}) + 3P(3\,\text{busy})\\
&= 1\binom{3}{1} P(1-P)^{2} + 2\binom{3}{2} P^{2}(1-P)\\
&\quad + 3\binom{3}{3} P^{3}(1-P)^{0}\\
&= 3P(1-P)^{2} + 6P^{2}(1-P) + 3P^{3}
\end{aligned}
$$

Solution 2.14.

$$
\begin{aligned}
&P \,\square\, \text{prob[a computer down]}\\
&\quad P(\text{at least one computer up in } a \text{ city})\\
&\quad = 1 - P(\text{both computers in } a \text{ city down}) \;]\\
&\quad = 1 - P^{2}\\
&\quad P(\text{at least 1 computer up in } \textit{each} \text{ city})\\
&\quad = (1-P^{2})^{3}
\end{aligned}
$$

Solution 2.15. No value appears to be feasible to reach 0.6 but in general one iterates on a calculator the formula:

$$
\begin{aligned}
&P(4 \text{ packets in } 10 \text{ slots})\\
&\quad = \binom{10}{4} P^{4}(1-P)^{6}\\
&\quad = 210P^{4}(1-P)^{6}
\end{aligned}
$$

Solution 2.16. For a binomial distribution the moment generating function is

$$
P(Z) = (ZP + 1 = P)^{N}
$$

A useful fact is

$$\text{2nd derivative}$$

$$E[i^2] = E^{(2)}[1] + E[i]$$

Then

$$E^{(2)}[1] = N(N-1)(ZP+1-P)^{N-2}P^2 |_{Z=1}$$

$$E^{(2)}[1] = N(N-1)P^2$$

So

$$E[i]$$

$$E[i^2] = N(N-1)P^2 + N_P$$

But

$$E[i]$$

$$\sigma^2 = E[i^2] - \mu^2$$

$$\sigma^2 = N(N-1)P^2 + NP - (NP)^2$$

$$\sigma^2 = NP(1-P)$$

Solution 2.17. Use a Pascal distribution:

$$P_N(K) = \begin{pmatrix} N & -1 \\ K & -1 \end{pmatrix} P^K (1-P)^{N-K}$$

$$= \begin{pmatrix} 9 \\ 4 \end{pmatrix} P^5 (1-P)^5$$

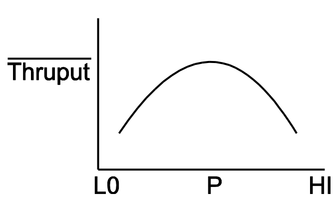
$$= \begin{pmatrix} 9 \\ 4 \end{pmatrix} (0.35)^5 (0.65)^5$$

$$= 126 \times (0.35)^5 (0.65)^5 = 0.07678$$

Solution 2.18.

$$\mu = \frac{K}{P} = \frac{5}{0.35} = 14.3$$

Solution 2.19.



mean thruput is maximized at a middle value, to the left load is light, to the right there are many collisions

Solution 2.20.

$$\text{Thruput} = NP(1-P)^{N-1}$$

$$\frac{d\ \text{Thruput}}{dP} = \frac{d}{dP}\left(NP(1-P)^{N-1}\right)$$

$$\cancel{N}P(N-1)(1-P)^{N-2}(-1) + \cancel{N}(1-P)^{N-1} = 0$$

$$-P(N-1)+(1-P) = 0$$

$$(1-P) = P(N-1)$$

$$1-P = PN - P$$

$$\boxed{P = \frac{1}{N}}$$

Solution 2.21. Because there is less contention, a station can acquire the channel sooner.

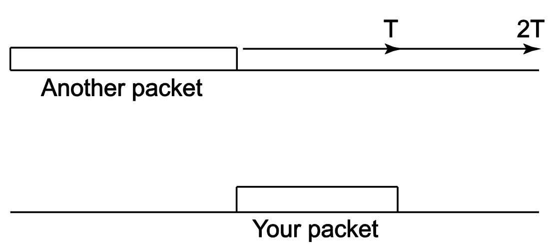
Solution 2.22.

$$U = \frac{1}{1+\frac{2BLe}{cF}} = \frac{1}{1+\frac{2\times 100\times 10^{6}\cdot 50\times 2.71}{3\times 10^{8}\times 512}}$$

$$= \frac{1}{1+0.176} = 0.85 = 85\%$$

Solution 2.23. For Fast Ethernet a 1 Gbps Ethernet network maximum size is already about 50 m. Since the size cannot be reduced further, it is better to either increase the minimum frame size, or use a switched hub rather than a shared media hub.

Solution 2.24.



Another packet can be anywhere in a "window" of 2T and overlap the packet of interest, causing a collision

Solution 2.25. Slotted Aloha:

$$S = Ge^{-G}$$

$$\frac{dS}{dG} = e^{-G} - Ge^{-G} = 0$$

$$1-G = 0$$

$$G = 1$$

$$S = 1\times e^{-1} = 0.368$$

Pure Aloha:

$$S = Ge^{-2G}$$
$$\frac{dS}{dG} = e^{-2G} - 2Ge^{-2G} = 0$$
$$1 - 2G = 0$$
$$G = \frac{1}{2}$$
$$S = \frac{1}{2}e^{-2\frac{1}{2}} = 0.184$$

Solution 2.26. Pure Aloha has a larger delay for large G because of its e^{2G} term, versus slotted Aloha's e^{G} term.

Solution 2.27. The three terms multiplied to form the terms in Eqs. (2.164) through (2.166) are (1) the number of arriving packets successfully going to the outputs in a slot weighted by (2) the probability that a certain pattern of packets prefers to go to each of the output ports in a slot for a given number of switching element arrivals and (3) the probability of a given number of arrivals in a slot.

Solution 2.28. In Fig. 2.19 there are two arrivals (A and B) and two outputs. Each instance of two connected squares represents two outputs. The placement of A and B indicates which output each of two arriving packets prefers (should be routed to). For instance, in the upper left boxes both packets A and B prefer the first output. In the lower right boxes packet B prefers output 1 and A packet prefers output 2.

Figure 2.20 is similar but is for three arriving packets. For instance, in the upper left boxes packets A, B, and C all prefer output 1. In the boxes immediately below that packet A prefers output 1 and packets B and C prefer output 2.

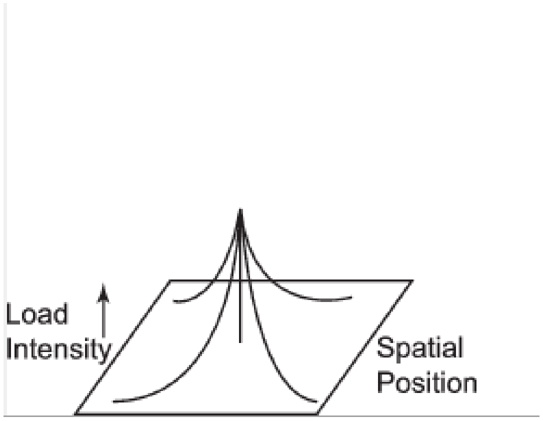
Solution 2.29. $\text{Load}_x = 2^{2\ell-1} - 2^{\ell-1}$

ℓ	$2\ell-1$	$\ell-1$	Load_x
2	3	1	6
5	9	4	496
10	19	9	523, 776

Solution 2.30. This is left as an exercise to the reader. Note that there are two components: one is traffic From/To nodes below the link and the other component is traffic between nodes below the

link going to nodes on the other side of the link. Be careful on the indexing.

Solution 2.31. Each bus interface in the knockout switch can be justified forwarding at most L packets since, if the switch parameters are chosen appropriately, the probability of more than L packets arriving is extremely small. Excess packets can be retransmitted by a higher level protocol if that is necessary. For some types of traffic it may not be.



Solution 2.32.

$$P\begin{pmatrix}\text{station xmts on}\\ \text{a given bus}\end{pmatrix} = \frac{P}{M}$$

$$\text{Thruput one bus} = \binom{N}{1}\left(\frac{P}{M}\right)^{1}\left(1-\frac{P}{M}\right)^{N-1}$$

$$\text{Thruput system} = MN\frac{P}{M}\left(1-\frac{P}{M}\right)^{N-1}$$

$$= NP\left(1-\frac{P}{M}\right)^{N-1}$$

N station

M buses

$R = M$

where R is # of connections to different buses per station.

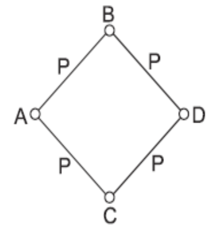
Solution 2.33. P: Probability a link is available

(a)

$$q = P^2$$

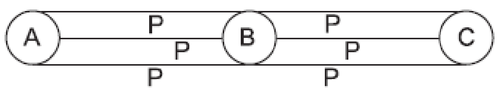
(b)

$$P\begin{bmatrix}\text{at least}\\ \text{one A to D path}\\ \text{available}\end{bmatrix} = 1 - P\begin{bmatrix}\text{no AD}\\ \text{path available}\end{bmatrix}$$



$$= 1 - (1 - P^2)^2$$

Solution 2.34.



P: probability a link is in use

(a)

$$\overline{\text{\# Busy Links}} = \sum_{n=1}^{N} n\binom{N}{n} P^n (1-P)^{N-n} = NP$$

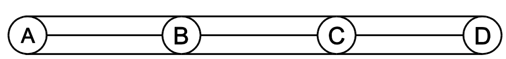
(b)

$$P\begin{bmatrix}\text{at least one}\\ \text{idle path from}\\ \text{A to C}\end{bmatrix} = \left(P\left(\begin{matrix}\text{at least 1 idle}\\ \text{link from A to B}\end{matrix}\right)\right)^2$$

$$= \left(1 - P\left(\begin{matrix}\text{no idle links}\\ \text{from A to B}\end{matrix}\right)\right)^2$$

$$= (1 - P^3)^2$$

Solution 2.35.



(a) P: Probability a link is busy

$$P\begin{pmatrix}\text{A to D}\\ \text{call blocked}\end{pmatrix} = P\begin{pmatrix}\text{at least one adjacent pair}\\ \text{is completely blocked}\end{pmatrix}$$
$$= 1 - P\begin{pmatrix}\text{at least one idle link}\\ \text{between each adjacent pair}\end{pmatrix}$$
$$= 1 - \left(P\begin{pmatrix}\text{at least one idle link between}\\ \text{an adjacent pair of nodes}\end{pmatrix}\right)^3$$
$$= 1 - (1 - P^3)^3$$

(b)

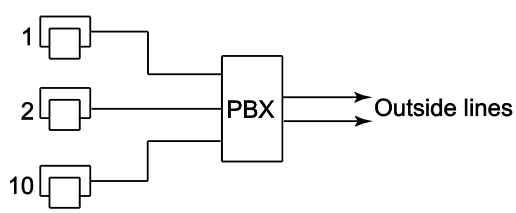
$$P\begin{pmatrix}\text{at least}\\ \text{two paths (available)}\\ \text{from A to D}\end{pmatrix} = P\begin{pmatrix}\text{at least 2}\\ \text{available links between}\\ \text{every adjacent pair of nodes}\end{pmatrix}$$
$$= \underbrace{(3(1-P)^2 P}_{\text{2 links available}} + \underbrace{(1-P)^3)^3}_{\text{3 links available}}$$

Solution 2.36. P: Probability a phone seeks an outside line.

(a)

$$P\begin{pmatrix}\text{n phones}\\ \text{seek an}\\ \text{outside line}\end{pmatrix} = \binom{10}{n} P^n (1-P)^{10-n}$$

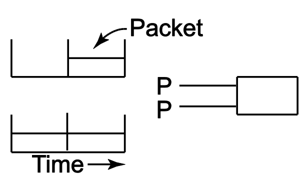
↗ binomial distribution



(b)

$$\text{P(blocking)} = P(> 2 \text{ phones seek an outside line})$$
$$= \sum_{n=3}^{10} \binom{10}{n} P^n (1-P)^{10-n}$$

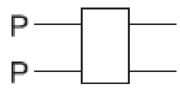
Solution 2.37.



$$P\begin{pmatrix}\text{3 or more packets}\\ \text{in 2 consecutive time slots}\end{pmatrix}=\begin{pmatrix}4\\3\end{pmatrix}P^3(1-P)+\begin{pmatrix}4\\4\end{pmatrix}P^4(1-P)^0$$

$$=\begin{pmatrix}4\\3\end{pmatrix}P^3(1-P)+P^4$$

Solution 2.38.



(a)

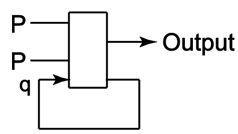
$$P\begin{pmatrix}\text{Packet on}\\ \text{specific}\\ \text{output}\end{pmatrix}=.5\overbrace{(2P(1-P))}^{\text{one arrival}}+\overset{\text{two arrivals}}{P^2}$$

$$\begin{pmatrix}\text{Single packet goes to}\\ \text{specific output}\end{pmatrix}$$

(b)

$$\overline{\text{\# packets at outputs}}=1-2\cdot P(1-P)+2\cdot P^2$$

Solution 2.39.



(a)

$$q=P\begin{pmatrix}\text{2 or more packets}\\ \text{at inputs}\end{pmatrix}$$

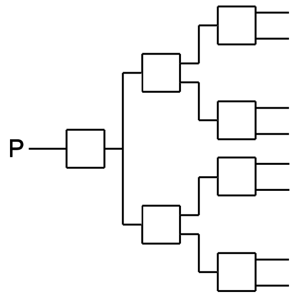
$$q=2P(1-P)q+P^2(1-q)+P^2q$$

$$q=2P(1-P)q+P^2$$

(b)

$$q=\frac{P^2}{1-2P(1-P)}$$

Solution 2.40.



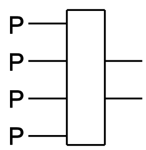
C: Probability a packet appears at switching element output if there is an input packet.

(a) For a given output, the probability a copy appears is C^3P.

(b)

$$\overline{\#\text{ Copies}} \cong \sum_{n=1}^{8} n \binom{8}{n} (C^3P)^n (1-C^3P)^{8-n}$$

Solution 2.41.

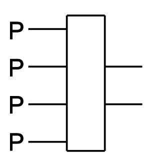


(a) $P(n\text{ arrivals}) = \binom{4}{n} P^n (1-P)^{4-n}$

(b)

$$\begin{aligned}(\text{at least 1 arrival}) &= 1 - P(0\text{ arrivals}) \\ &= 1 - \binom{4}{0} P^0 (1-P)^{4-0} \\ &= 1-(1-P)^4\end{aligned}$$

Solution 2.42.



(a)

$$\begin{aligned}P\begin{pmatrix}\text{packet dropped} \\ \text{(bird eye's view)}\end{pmatrix} &= P(3\text{ or more arrivals}) \\ &= \binom{4}{3} P^3 (1-P) + \binom{4}{4} P^4 (1-P)^0\end{aligned}$$

(b)

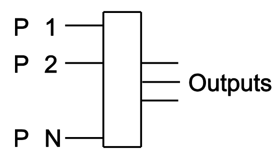
$$P\begin{pmatrix}\underline{\text{Tagged}}\text{ arriving}\\ \text{packet dropped}\end{pmatrix}=\frac{1}{3}P(2\ \underline{\text{other}}\text{ packets arrive})+\frac{1}{2}P\begin{pmatrix}3\ \underline{\text{other}}\\ \text{packets}\\ \text{arrive}\end{pmatrix}$$

$$=\frac{1}{3}\binom{3}{2}P^2(1-P)+\binom{3}{3}P^3(1-P)^0$$

(c)

$$\overline{\text{Thruput}}=1\cdot P(1\text{ arrival})+2P(2\text{ or more arrivals})$$
$$=1\binom{4}{1}P(1-P)^3+2\sum_{n=2}^{4}\binom{4}{n}P^n(1-P)^{4-n}$$

Solution 2.43.



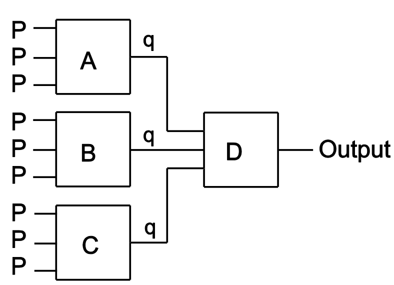
(a)

$$\overline{\text{Thruput}}=1\cdot P(1\text{ arrival})+2P(2\text{ arrivals})+3P(3\text{ or more arrivals})$$
$$=1\binom{N}{1}P(1-P)^{N-1}+2\binom{N}{2}P^2(1-P)^{N-2}$$
$$+3\sum_{n=3}^{N}\binom{N}{n}P^n(1-P)^{N-n}$$

(b)

$$\overline{\#\text{ Dropped packets}}=\sum_{n=4}^{N}(n-3)\binom{N}{n}P^n(1-P)^{N-n}$$

Solution 2.44.



(a) For either A, B, or C

$$q = P(1\text{ or more arrivals to element})$$
$$=1-(1-P)^3$$

(b)

$$\overline{\text{Thruput}} = P(\text{1 or more arrivals to D})$$
$$= 1-(1-q)^3$$

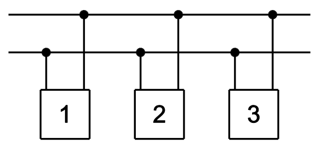
Solution 2.45. $\dfrac{NR}{m}$ should be integer:

$N \to \#$ stations

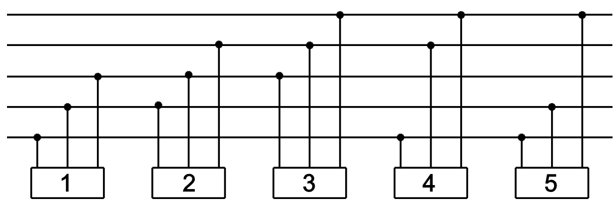
$M \to \#$ buses

$R \to \#$ connections/stations

(a) $N = 3 \quad R = 2 \quad M = 2$



(b) $N = 5 \quad R = 3 \quad M = 5$



3

Queueing Models

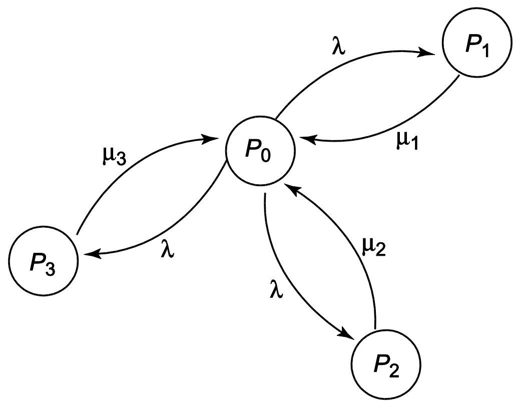
1. In an M/M/1 queue the arrival process is Poisson and the service time distribution is (independent) negative exponential. In a Geom/Geom/l queue the arrival process is Bernoulli and a customer in the server completes service in each slot with independent probability p (i.e., service times follow a geometric distribution).
2. To say the M/M/1 queue is memoryless means that the queue's state is completely summarized by the current number of customers in the queue. There is no need to include the time since the last arrival or the time the current customer in the server has been in service. A knowledge of this past history has no increased predictive power. This is because the negative exponential distribution which underlies the arrival (Poisson) and service processes is "memoryless." It is the only continuous time memoryless distribution. The geometric distribution is the only discrete time memoryless distribution.
3. Electric current flows in electric circuits. Probability flux flows in Markov chains. The

probability flux flowing on a link is the average number of times a second the system state transits the transition. Current can flow in either direction in a simple resistive electric circuit. Current flow direction is preset by transition direction in a Markov chain. Voltage/current sources guarantee non-zero flows of current in electric circuits. The normalization equation guarantees non-zero flows of probability flux in Markov chains.

4. Using Little's law for a packet buffer, either mean thruput, mean delay, or mean number of packets in the buffer can be found simply using the formula from the other two quantities.
5. Global balance holds that in equilibrium the total flow of probability flux into a Markov chain state equals the total flow of probability flux out of the same Markov chain state. The "total" from is the flow summed over all incoming/outgoing transitions to/from a state. Global balance applies to any Markov chain in equilibrium. Local balance only applies to certain Markov chains. Under local balance the flow in and out of certain subsets of transitions to/from a state balances. Local balance is a characteristic of Markov chains associated with product from queueing or Petri nets.
6. Global balance equations are usually solved using linear equations with one equation per state. Unfortunately the number of states often goes up exponentially with the size of a system. Moreover the computational complexity of general linear equation solution is proportional to the cube of the number of equations. Thus the computational burden can be overwhelming.
7. From Burke's theorem, if an M/M/1 queue input follows a Poisson process, the output process is also Poisson.
8. Yes, the arrival rate can be greater for an M/M/1 finite buffer queue as all this means that the buffer is often full and many customers are turned away. By way of contrast, for an infinite buffer M/M/1 queue if the arrival rate is greater than the service rate, then the queue is unstable (the waiting line continually increases in size).
9. The Erlang B formula and Erlang C formula both involve a Poisson input of calls and negative exponential service time distributions for each parallel server. However the Erlang B formula is appropriate when there is no waiting line so that if an incoming call does not immediately find an idle server, it is dropped. The Erlang C formula applies to systems with a queue so an incoming call is queued if it does not immediately find an idle server until it is at the head of the line of the queue and an idle server becomes available.

10. The queueing based memory model is appropriate for computer systems where incoming jobs have memory requirements and memory is a shared resource.
11. A Markov chain imbedded at the departure instants is the system Markov chain that exists only at departure instants.
12. For queueing network product form results we assume Poisson inputs (for an open network), negative exponential servers, and (independent) random routing.
13. Traffic equations model the mean throughput (flow) of customers through a queueing network.
14. The mean value analysis algorithm is based on Little's Law. It is also based on the insight that an arriving customer in a closed Markovian queueing network "sees" a number of customers that follows the equilibrium distribution with one less customer in the network.
15. When a negative customer enters a queue with a least one positive customer, the negative customer and one of the positive customers are instantly removed from the network. When a negative customer enters an empty queue, the negative customer is instantly removed from the queue and dropped from the system.
16. Non-product networks can in certain cases be solved with simple recursions. In general numerical linear equations are necessary.
17. Petri networks model concurrency, serializability, synchronization, and resource sharing.
18. A specific marking (placement of tokens to places) is a state.
19. Analytical results, though sometimes difficult to obtain, are usually simple to solve and give insight into performance tradeoffs.
20. Simulation results will not always match experimental results because there may be real world factors that influence an experiment that is not programmed into a particular simulation. A simulation will not indicate anything beyond its programming.
21. Markovian queueing networks, Markovian negative customer queueing networks, and certain stochastic Petri nets have product form solutions.

Solution 3.22.



(a) Using a boundary across each arm:

$$\mu_1 P_1 = \lambda P_0 \qquad P_1 = \frac{\lambda}{\mu_1} P_0 \qquad P_0 + P_1 + P_2 + P_3 = 1$$

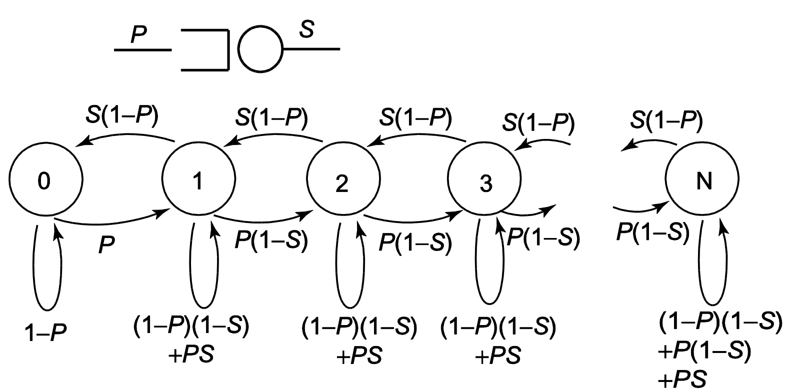
$$\mu_2 P_2 = \lambda P_0 \quad \Rightarrow \quad P_2 = \frac{\lambda}{\mu_2} P_0 \quad \Rightarrow \quad P_0 + \frac{\lambda}{\mu_1} P_0 + \frac{\lambda}{\mu_2} P_0 + \frac{\lambda}{\mu_3} P_0 = 1$$

$$\mu_3 P_3 = \lambda P_0 \qquad P_3 + \frac{\lambda}{\mu_3} P_0 \qquad P_0 = \frac{1}{1 + \frac{\lambda}{\mu_1} + \frac{\lambda}{\mu_2} + \frac{\lambda}{\mu \mathrm{s}}}$$

(b) $P_B = 1 - P_0 =$ Utilization

Solution 3.23.

(a)



(b)

$$n = 1: \qquad P_n = \frac{P}{S(1-P)} P_0$$

$$n = 2, 3 \ldots N: \quad P_n = \frac{P}{S(1-P)} \left(\frac{P(1-S)}{S(1-P)} \right)^{n-1} P_0$$

$$n = 1, 2 \ldots N: \quad P_n = \frac{P^n (1-S)^{n-1}}{S^n (1-P)^n} P_0$$

(c)

$$P_0 = \frac{1}{\sum_{i=1}^{\infty} \frac{P^i (1-S)^{i-1}}{S^i (1-P)^i} + 1} \quad \text{from normalization equation}$$

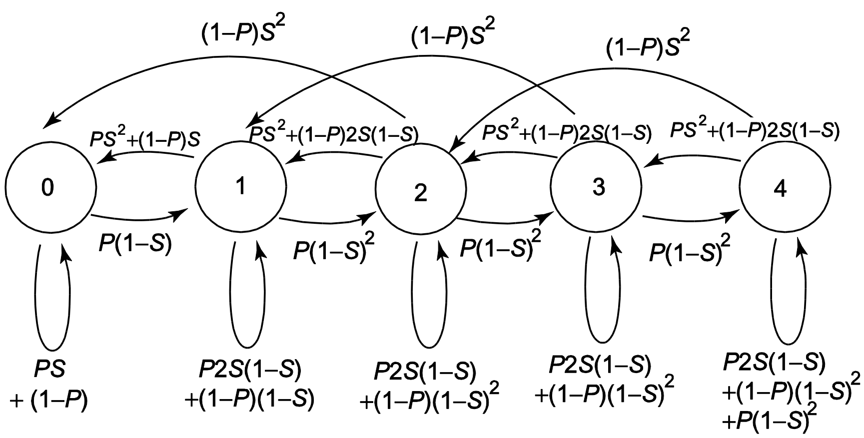
$$P_0 = \frac{(1-S)}{\sum_{i=1}^{\infty} \frac{P^i (1-S)^i}{S^i (1-P)^i} + 1 - S}$$

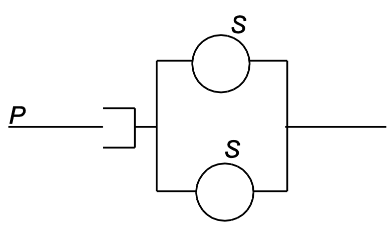
$$P_0 = \frac{(1-S)}{\sum_{i=0}^{\infty} \frac{P^i (1-S)^i}{S^i (1-P)^i} - S}$$

$$P_0 = \frac{1-S}{\frac{1}{1-\frac{P(1-\mathrm{S})}{\mathrm{S}(1-P)}} - S}$$

with algebra : $P_0 = 1 - \frac{P}{S}$

Solution 3.24.



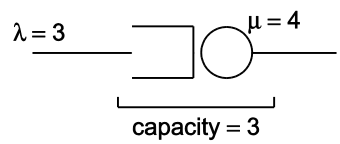


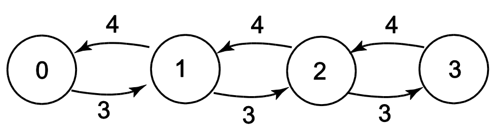
Solution 3.25.

n	$P(n)$
0	0.15
1	0.20
2	0.35
3	0.30

$$\bar{n} = 1\times 0.2 + 2\times 0.35 + 3\times 0.3$$
$$= 0.2 + 0.7 + 0.9 = 1.8$$
$$\overline{\text{Thruput}} = \mu(0.2 + 0.35 + 0.3) = 0.85\mu$$
$$\overline{\text{Delay}} = \frac{\bar{n}}{\overline{\text{Thruput}}} = \frac{1.8}{0.85\mu} = \frac{2.11}{\mu}$$

Solution 3.26.





(a)

$$P_1 = \frac{3}{4}P_0 \qquad P_0 + P_1 + P_2 + P_3 = 1$$
$$P_2 = \frac{3}{4}P_1 = \frac{9}{16}P_0 \qquad P_0 = \frac{1}{1 + \frac{3}{4} + \frac{9}{16} + \frac{27}{64}}$$
$$P_3 = \frac{3}{4}P_2 = \frac{27}{64}P_0 \qquad = \frac{1}{2.7343} = 0.36571$$
$$P_1 = 0.27428 \quad P_2 = 0.20571$$
$$P_3 = 0.15429$$

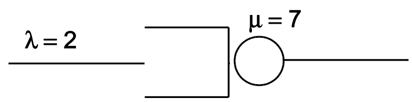
(b)

$$\overline{\text{Delay}} = \frac{\bar{n}}{\overline{\text{Thruput}}}$$
$$= \frac{1P_1 + 2P_2 + 3P_3}{4(P_1 + P_2 + P_3)} = 0.45270\,\text{s}$$

(c)

$$\overline{\text{Delay}}_{\text{buffer}} = \overline{\text{Delay}}_{\text{Tota1}} - \overline{\text{Delay}}_{\text{Server}}$$
$$= 0.45270 - \frac{1}{4} = 0.2027\,\text{s}$$

Solution 3.27.



(a) Infinite buffer

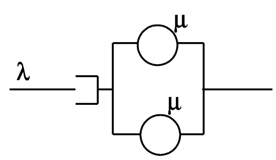
$$P_0 = 1 - \rho = 1 - \frac{2}{7} = 0.714$$
$$U = 1 - P_0 = \rho = 2/7 = 0.28571$$

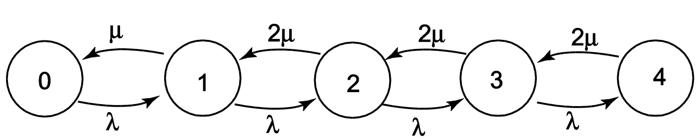
(b) Finite buffer $(N = 4)$

$$U = 1 - P_0$$
$$P_0 = \frac{1 - \frac{\lambda}{\mu}}{1 - \left(\frac{\lambda}{\mu}\right)^{N+1}}$$
$$= \frac{1 - \frac{2}{7}}{1 - \left(\frac{2}{7}\right)^5} = \frac{5/7}{0.998} = 0.71564$$
$$U = 1 - P_0 = 1 - 0.71564 = 0.28436$$

(c) The infinite buffer queue has a larger utilization as no customer is turned away.

Solution 3.28.





$$P_1 = \frac{\lambda}{\mu} P_0$$

$$P_2 = \frac{\lambda}{2\mu} P_1 = \frac{\lambda^2}{2\mu^2} P_0$$

$$P_3 = \frac{\lambda}{2\mu} \frac{\lambda^2}{2\mu^2} P_0 = \frac{\lambda^3}{4\mu^3} P_0$$

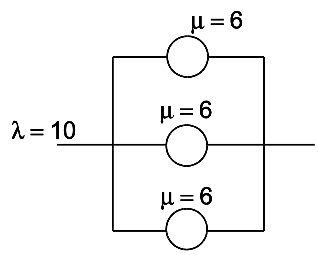
$$P_4 = \frac{\lambda}{2\mu} \frac{\lambda^3}{4\mu^3} P_0 = \frac{\lambda^4}{8\mu^4} P_0$$

$$P_0 + P_1 + P_2 + P_3 + P_4 = 1$$

$$P_0 = \frac{1}{1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{2\mu^2} + \frac{\lambda^3}{4\mu^3} + \frac{\lambda^4}{8\mu^4}}$$

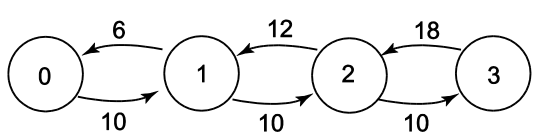
$$P_B = P_4 = \frac{\frac{1}{8}\left(\frac{\lambda}{4}\right)^4}{1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{2\mu^2} + \frac{\lambda^3}{4\mu^3} + \frac{\lambda^4}{8\mu^4}}$$

Solution 3.29.



(a) Avg # empty servers $= 3P_0 + 2P_1 + P_2$ where P_i is the equilibrium probability of i customers in the queueing system.

(b)



$$P_1 = \frac{10}{6} P_0$$

$$P_2 = \frac{10}{12} P_1 = \frac{100}{72} P_0$$

$$P_3 = \frac{10}{18} P_2 = \frac{1000}{1296} P_0$$

$$P_0 + P_1 + P_2 + P_3 = 1$$

$$P_0 = \frac{1}{1+\frac{10}{6}+\frac{100}{72}+\frac{1000}{1296}}$$

$$P_0 = 0.207161125$$

Avg # empty servers

$$= 3P_0 + 2P_1 + P_2$$
$$= 1.5997$$

Solution 3.30.

$$\mu = \frac{60}{12} = 5 \text{ calls/h}$$

$$\lambda = \frac{60}{9} = 6.66 \text{ calls/h}$$

(a) M/M/m, m parallel servers with a queue, blocked calls are queued.

(b) Erlang C

Solution 3.31.

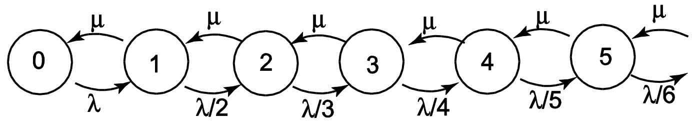
$$\text{Avg \# Customers in queue} = \frac{1}{2}\times 1 + \frac{1}{2}\times 0$$
$$= \frac{1}{2}$$

Solution 3.32. Since no customers are lost,

$$\text{Thruput} = \lambda,$$

Solution 3.33.

(a)



(b)

$$
\begin{aligned}
P_n &= \frac{\lambda^n}{n!\mu^n} P_0 \quad n = 1, 2, 3 \ldots \\
\text{as} \quad P_1 &= \frac{\lambda}{\mu} P_0 \\
P_2 &= \frac{\lambda}{2\mu} P_1 = \frac{\lambda^2}{2\mu^2} P_0 \\
P_3 &= \frac{\lambda}{3\mu} P_2 = \frac{\lambda^3}{6\mu^3} P_0 \\
&\vdots
\end{aligned}
$$

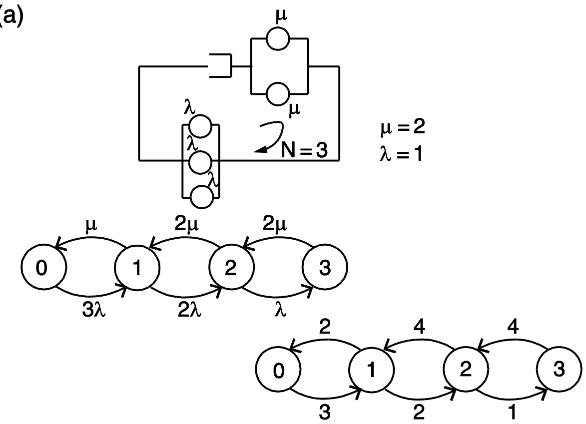
(c)

$$
\begin{aligned}
&P_0 + P_1 + P_2 + P_3 + \ldots = 1 \\
&P_0 \left(\sum_{n=0}^{\infty} \frac{\lambda^n}{n!\mu^n} \right) = 1 \\
&P_0 = \frac{1}{\sum_{n=0}^{\infty} \frac{\lambda^n}{n!\mu^n}} = \frac{1}{e^{\lambda/\mu}} = e^{-\lambda/\mu}
\end{aligned}
$$

from well-known summation formula.

Solution 3.34.

(a)



(b)

$$P_1 = \frac{3}{2}P_0$$

$$P_2 = \frac{1}{2}P_1 = \frac{3}{4}P_0$$

$$P_3 = \frac{1}{4}P_2 = \frac{3}{16}P_0$$

$$P_0 + P_1 + P_3 + P_3 = 1$$

$$P_0 + \frac{3}{2}P_0 + \frac{3}{4}P_0 + \frac{3}{16}P_0 = 1$$

$$P_0 = \frac{1}{1 + \frac{3}{2} + \frac{3}{4} + \frac{3}{16}} = \frac{16}{55}$$

(c)

Thruput upper queue = 2 x $P_1 + 4P_2 + 4P_3$

Solution 3.35.

$$P(\underline{n}) = \left(\prod_{i=1}^{k} \frac{a_i^{n_i}}{n_i!}\right) G^{-1}(\Omega)$$

$$P(\underline{n}) = \left(\frac{(1/2)^{n_1}}{n_1!}\right)\left(\frac{(1/3)^{n_2}}{n_2!}\right) \Big/ \sum_{\underline{n}\in\Omega} \left(\frac{(1/2)^{n_1}}{n_1!}\right)\left(\frac{(1/3)^{n_2}}{n_2!}\right)$$

$$\sum n_1 + 3n_2 \le 1000, \quad n_1, n_2 = \text{X M bytes}$$

Solution 3.36.

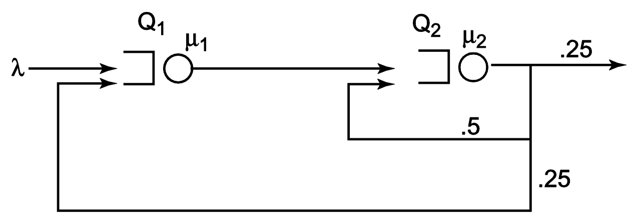
$$\text{M/D/1} \quad E[n] = \rho + \frac{\rho^2}{2(1-\rho)}$$

$$\text{M/M/1} \quad E[n] = \frac{\rho}{1-\rho}$$

ρ	M/M/1	M/D/1	– As ρ → 1
0.1	0.111	0.106	

<table>
<tr><td>0.2</td><td>0.250</td><td>0.225</td><td rowspan="6">$E[n] = 2 \times E[n]$
M/M/1 $M/D/1$
M/M/1 has larger average queue size because of longer (more variable) distribution tail</td></tr>
<tr><td>0.4</td><td>0.667</td><td>0.533</td></tr>
<tr><td>0.6</td><td>1.50</td><td>1.05</td></tr>
<tr><td>0.8</td><td>4.00</td><td>2.40</td></tr>
<tr><td>0.9</td><td>9.00</td><td>4.95</td></tr>
<tr><td>0.99</td><td>99.0</td><td>50.0</td></tr>
</table>

Solution 3.37. (a)



$$\left.\begin{aligned}\theta_1 &= \lambda + .25\theta_2 \\ \theta_2 &= \theta_1 + .5\theta_2\end{aligned}\right\}\text{Traffic equations}$$

$$\theta_2 = 2\theta_1$$

$$\downarrow$$

$$\theta_1 = \lambda + .5\theta_1$$

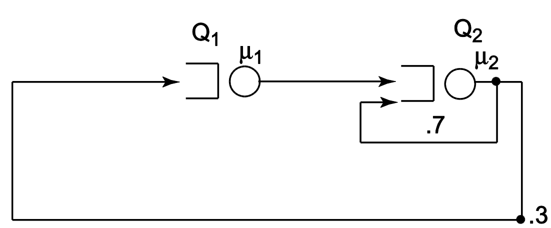
$$\boxed{\begin{aligned}\theta_1 &= 2\lambda \\ \theta_2 &= 4\lambda\end{aligned}}$$

(b)

$$P(\underline{n}) = \left(\frac{2\lambda}{\mu_1}\right)^{n_1}\left(\frac{4\lambda}{\mu_2}\right)^{n_2} P(\underline{0})$$

$$P(\underline{0}) = \frac{1}{\sum_{\underline{n}}\left(\left(\frac{2\lambda}{\mu_1}\right)^{n_1}\left(\frac{4\lambda}{\mu_2}\right)^{n_2}\right)}$$

Solution 3.38.



$$\left.\begin{aligned}\theta_1 &= .3\theta_2\\ \theta_2 &= \theta_1 + .7\theta_2\end{aligned}\right\}\text{Traffic equations}$$

$$\theta_2 = \frac{10}{3}\theta_1$$

so $\quad \theta_1 = 1, \theta_2 = 10/3$

so $\theta_1 = 1, \theta_2 = 10/3$

Solution 3.39. Global balance equation:

$$\sum_{i=1}^{M} \mu_i P(\underline{n}) = \sum_{i=1}^{M}\sum_{j=1}^{M} \mu_j r_{ji} P(\underline{n} + 1_j - 1_i)^*$$

Traffic equations:

$$\underline{1}_i = [0, 0, 0 \ldots 1, \ldots 0, 0]$$

*Here $\qquad\qquad \uparrow$

*i*th position

$$\theta_j = \sum_{j=1}^{M} r_{ji}\theta_j \quad i = 1, 2, M$$

Manipulate the traffic equations:

$$1 = \sum_{j=1}^{M} r_{ji}\frac{\theta_j}{\theta_i} \quad i = 1, 2, \ldots M$$

Multiply the left-hand side of the global balance equation by this expression for one:

$$\sum_{i=1}^{M}\sum_{j=1}^{M} \mu_i r_{ji}\frac{\theta_j}{\theta_i} P(\underline{n})$$
$$= \sum_{i=1}^{M}\sum_{j=1}^{M} \mu_j r_{ji} P(\underline{n} + \underline{1}_j - \underline{1}_i)$$

Rearrange as

$$\sum_{i=1}^{M}\sum_{j=1}^{M} r_{ji}\left(\frac{\theta_j}{\theta_i}\mu_i P(\underline{n}) - \mu_j P(\underline{n} + \underline{1}_j - \underline{1}_i)\right) = 0$$

If the parenthesis term is 0, this equation is satisfied when

$$\frac{\theta_j}{\theta_i}\mu_i P(\underline{n}) = \mu_j P(\underline{n}+\underline{1}_j-\underline{1}_i) \quad i,\ j=1,2,\ldots M$$

Manipulating:

$$P(\underline{n}) = \left(\frac{\theta_i}{\mu_i}\right)\left(\frac{\theta_j}{\mu_j}\right)^{-1} P(\underline{n}+\underline{1}_j-\underline{1}_i) \quad i,\ j=1,2,\ldots M$$

which yields

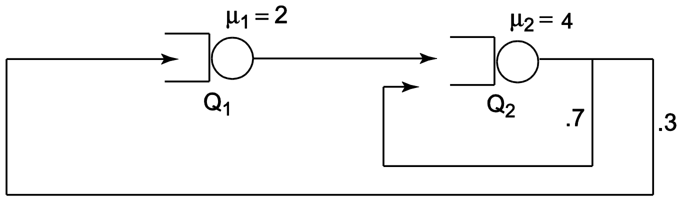
$$P(\underline{n}) = \left(\frac{\theta_i}{\mu_i}\right)P(\underline{n}-\underline{1}_i)$$

or

$$\mu_i P(\underline{n}) = \theta_i P(\underline{n}-\underline{1}_i)$$

which is the desired result.

Solution 3.40. Refering to problem 3.38:



From 3.38, $\theta_2 = \frac{10}{3}\theta_1$ or $\theta_1 = 1 \quad \theta_2 = 10/3$

$\underline{N=1}$

$$\bar{\tau}_1(1) = 0.5$$
$$\bar{\tau}_2(1) = 0.25$$

$$\bar{T}(1) = \frac{1}{1+0.5+\frac{10}{3}\times 0.25} = 0.75$$

$$\bar{n}_1(1) = 0.75\times 1\times 0.5 = 0.375$$

$$\bar{n}_2(1) = 0.75\times\frac{10}{3}\times 0.25 = 0.625$$

$\underline{N=2}$

$$\bar{\tau}_1(2) = 0.5+0.5\times 0.375 = 0.6875$$
$$\bar{\tau}_2(2) = 0.25+0.25\times 0.625 = 0.40625$$

$$\bar{T}(2) = \frac{2}{1 \times 0.6875 + \frac{10}{3} \times 0.40625} = 0.9796$$

$$\bar{n}_1(2) = 0.9796 \times 1 \times 0.6875 = 0.6735$$

$$\bar{n}_2(2) = 0.9796 \times \frac{10}{3} \times 0.40625 = 1.327$$

$\underline{N = 3}$

$$\bar{\tau}_1(3) = 0.5 + 0.5 \times 0.6735 = 0.83675$$

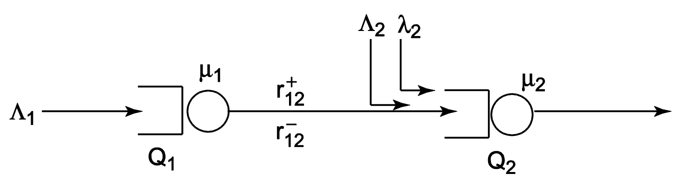
$$\bar{\tau}_2(3) = 0.25 + 0.25 \times 1.327 = 0.58175$$

$$\bar{T}(3) = \frac{3}{1 \times 0.83675 + \frac{10}{3} \times 0.58175} = 1.081$$

$$\bar{n}_1(3) = 1.081 \times 1.0 \times 0.83675 = 0.905$$

$$\bar{n}_2(3) = 1.081 \times \frac{10}{3} \times 0.58175 = 2.096$$

Solution 3.41.



$$q_1 = \frac{\wedge_1}{\mu_1}$$

$$q_2 = \frac{\wedge_2 + q_1\mu_1 r_{12}^+}{\mu_2 + q_1\mu_1 r_{12}^- + \lambda_2}$$

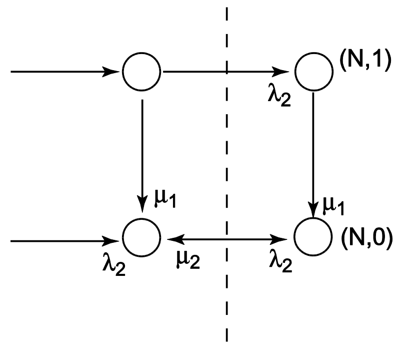
Or:

$$q_2 = \frac{\wedge_2 + \wedge_1 r_{12}^+}{\mu_2 + \wedge_1 r_{12}^- + \lambda_2}$$

And:

$$P(\underline{n}) = \prod_{i=1}^{2} (1 - q_i) q_i^{n_i}$$

Solution 3.42. For e.g. (3.250):



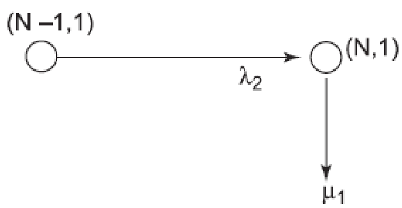
Balancing flow across the boundary,

$$\lambda_2(P(N-1,0)+P(N-1,1)) = \mu_2 P(N,0)$$

Leading to

$$P(N,0) = \frac{\lambda_2}{\mu_2}[P(N-1,0)+P(N-1,1)]$$

For example, (3.251)

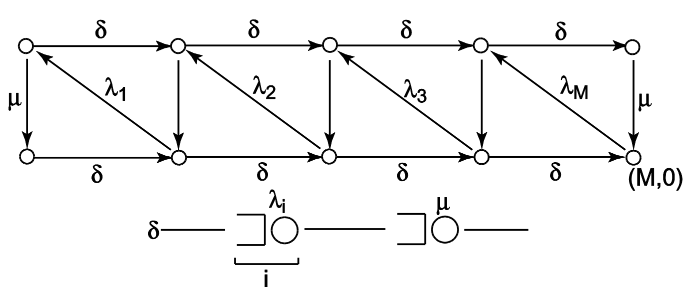


A global balance, for example, at $(N,1)$ yields

$$\lambda_2 P(N-1,1) = \mu_1 P(N,1)$$

$$P(N,1) = \frac{\lambda_2}{\mu_1}(P(N-1,1))$$

Solution 3.43. (a)



(b) Initially:

$$p(0,0) = 1.0 \quad p(0,1) = \left(\frac{\delta}{\mu}\right) p(0,0)$$

Recursively $i = 1, 2, \ldots M - 1$

$$p(i, 0) = \left(\frac{\delta}{\lambda_i}\right)[p(i-1, 0) + p(i-1, 1)]$$

$$p(i, 1) = \left(\frac{\delta}{\mu}\right)[p(i-1, 1) + p(i, 0)]$$

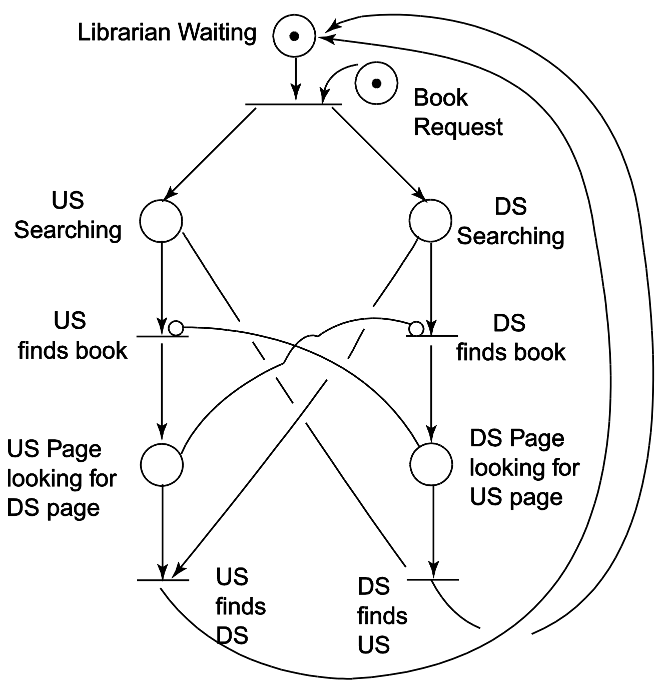
At the (right) boundary:

$$p(M, 0) = \left(\frac{\delta}{\lambda_M}\right)[P(M-1, 0) + P(M-1, 1)]$$

$$P(M, 1) = \left(\frac{\delta}{\mu}\right)P(M-1, 1)$$

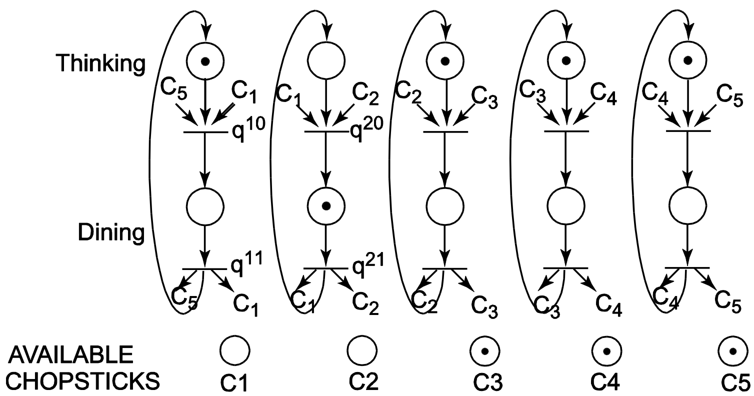
Solution 3.44.

US : UPSTAIRS PAGE DS : DOWNSTAIRS PAGE

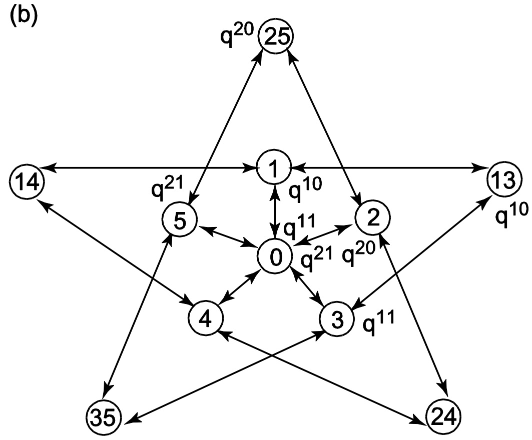


Solution 3.45.

(a)

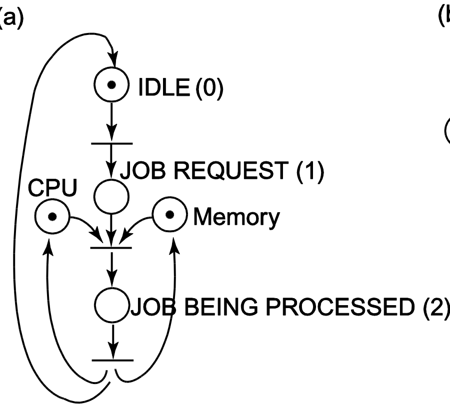


(b)

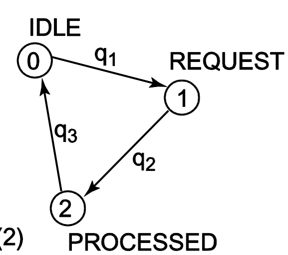


Solution 3.46.

(a)



(b)



$$q_1 P_{\text{IDLE}} = q_3 P_{\text{PROCESSED}}$$

$$P_{\text{PROCESSED}} = \frac{q_1}{q\text{s}} P_{\text{IDLE}}$$

$$\boxed{P_{\text{JOB REQUEST}} = \frac{q_1}{q_2} P_{\text{IDLE}}}$$

Likewise:

$$\left(1 + \frac{q_1}{q_3} + \frac{q_1}{q_2}\right) P_{\text{IDLE}} = 1$$

$$P_{\text{IDLE}} = \frac{1}{1 + \frac{q_1}{q_3} + \frac{q_1}{q_2}} = \frac{q_2 q_3}{q_2 q_3 + q_1 q_2 + q_1 q_3}$$

$$P_{\text{PROCESSED}} = \rightarrow = \frac{q_1 q_2}{q_1 q_3 + q_1 q_2 + q_1 q_3}$$

$$P_{\text{JOB REQUEST}} = \rightarrow = \frac{q_1 q_3}{q_2 q_3 + q_1 q_2 + q_1 q_3}$$

4

Fundamental Deterministic Algorithms

1. See below.
2. The Dijkstra and Ford Fulkerson algorithms each find all the shortest paths from a root node to all of the other nodes in the network in one run.
3. To find the shortest paths between every pair of nodes in a network, one runs either the Dijkstra or Ford Fulkerson algorithm N times, each time using a different node as the root node.
4. To find the k shortest link disjoint paths between a pair of nodes one first runs a shortest path algorithm to find the shortest path between the pair of nodes. Links along this path are then removed from the network. Again, one runs a shortest path algorithm, finds the shortest path between the two nodes, and removes the links from this path from the network. One does this k times or until the remaining network is partitioned between the two nodes.
5. The “algorithm table” is used in the calculation of shortest paths and contains intermediate as well as final results. A “routing table” is generated from the algorithm table and is placed in each node. It allows the output port to the next neighbor along a path be looked up for an outgoing packet or circuit.
6. See below.
7. A typical way to find routes under source routing is to flood discovery packets that maintain

a record of the routes they take. When a discovery packet(s) arrives at the destination, the destination can extract the path and send a packet back to the source with the route. If several discovery packets arrive at a destination, the destination can select the "best" route according to some metric.

8. Under pure flooding packets are sent out from a node in all directions. A node receiving a flooded packet forwards it on all of its outgoing links except the link it arrived on. The number of packets generated under flooding can be reduced if there is some sense of direction so packets are only forwarded in directions of interest or directions away from their point of generation.
9. Paths between nodes in a hierarchical network are sometimes longer than direct connections because they are not necessarily shortest paths.
10. The concept of switching elements is useful in VLSI design because it is easy with VLSI to replicate simple switching element building blocks many times on a chip.
11. A disadvantage of putting multiple addresses into packet for multicasting is the packet length increase (though there are less total packets) and the need for software modifications to the nodes. A disadvantage of the use of spanning trees for multicasting is the need for software modifications to the nodes.
12. Because it takes more energy in a nonlinear way to go further distances (usually according to some power law), a simple calculation shows the total energy consumed in a series of short hops is significantly smaller than the energy consumed in one large hop with equivalent total distance.
13. Reactive routing in ad hoc networks is more efficient than proactive routing when only a small fraction of potential paths are in use as under reactive routing information is only maintained for paths in use.
14. The complexity of protocol verification for large systems increases combinatorially. This necessitates the use of efficient protocol verification algorithms.
15. Under a "deadlock" a system freezes and cannot move into a further state. Under "livelock" messages are continually transmitted and received with no useful work accomplished.
16. Under an unspecified reception a message in the channel may be received but not as initially specified in the design.
17. The bits in four positions differ between 00001111 and 11001100 so the Hamming distance

between the two code words is four.

18. In Fig. 4.10, a legitimate codeword can be conceivably corrupted in such a way that it becomes a legitimate codeword in a different region and so the error is not detected. This occurs either with low probability or with types of errors that are outside the design of the code.
19. Yes, try a few examples to see this.
20. If the codeword "250" is received, the error is not detected as the codeword is divisible by the generator number 25. See problem 4.18.

Solution 4.1.

Any shortest path from A to Z has four Ups (U) and nine Rights (R)

$$4 \text{ shortest paths} = \binom{9+4}{4} = \binom{9+4}{9} = \binom{13}{9} = 715 \text{ paths.}$$

Solution 4.6. NODE C Routing Table

Destination	Nearest neighbor
A	F
B	B
D	E
E	E
F	F

Solution 4.21.

Dijkstra Algorithm						
N	B	C	D	E	F	G
$\{A\}$	2	∞	10	∞	10	∞
$\{A, B\}$	2	3	10	∞	10	∞
$\{A-C\}$	2	3	10	4	10	4

$\{A-C, E\}$	2	3	5	4	10	4
$\{A-C, E, G\}$	2	3	5	4	5	4
$\{A-E, G\}$	2	3	5	4	5	4
$\{A-G\}$	2	3	5	4	5	4

Ford-Fulkerson Algorithm						
	B	C	D	E	F	G
Initialize	$(\cdot,\infty)$	$(\cdot,\infty)$	$(\cdot,\infty)$	$(\cdot,\infty)$	$(\cdot,\infty)$	$(\cdot,\infty)$
1	$(A,2)$	$(B,3)$	$(A,10)$	$(C,4)$	$(A,10)$	$(C,4)$
2	$(A,2)$	$(B,3)$	$(E,5)$	$(C,4)$	$(G,5)$	$(C,4)$
3	$(A,2)$	$(B,3)$	$(E,5)$	$(C,4)$	$(G,5)$	$(C,4)$

Bottleneck Routing Algorithm

	N	B	C	D	E	F	G
1.	{A}	(A, 2)	(–, 0)	(A, 10)	(–, 0)	(A, 10)	(–, 0)
2.	{A, D}	(A, 2)	(–, 0)	*(A, 10)*	(D, 1)	(A, 10)	(–, 0)
3.	{A, D, F}	(A, 2)	(–, 0)	(A, 10)	(D, 1)	*(A, 10)*	(F,1)
4.	{A, B, D, F}	*(A, 2)*	(B, 1)	(A, 10)	(D, 1)	(A, 10)	(F,1)
5.	{A, B, C, D, F}	(A, 2)	*(B, 1)*	(A, 10)	(D, 1)	(A, 10)	(F,1)
6.	{A, F}	(A, 2)	(B, 1)	(A, 10)	*(D, 1)*	(A, 10)	(F, 1)
7.	{A, 6}	(A, 2)	(B, 1)	(A, 10)	(D, 1)	(A, 10)	*(F, 1)*

Solution 4.22.

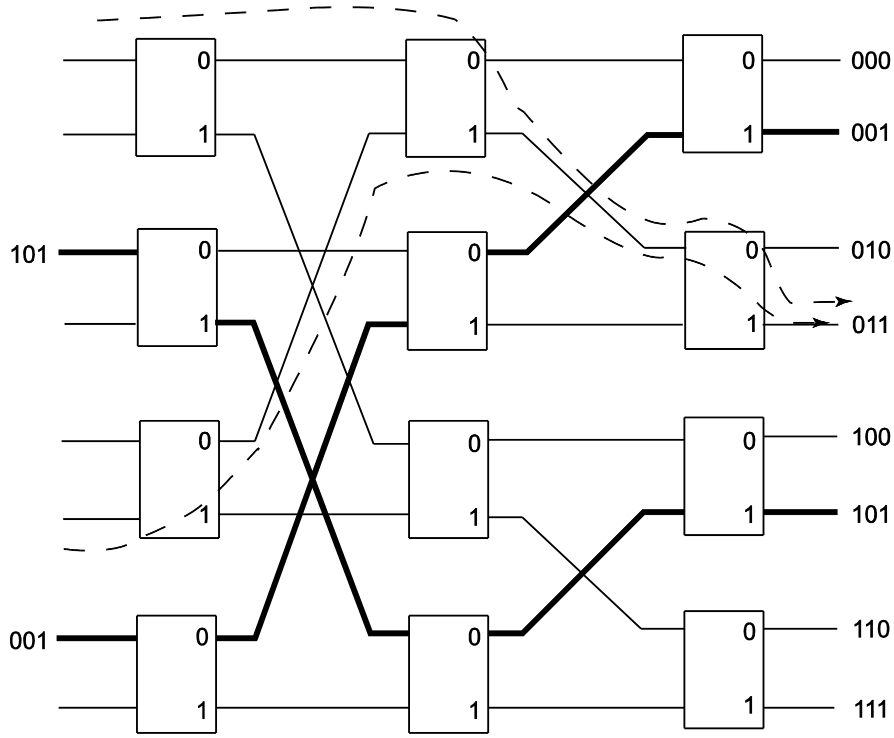


Fig. 4.4. An 8×8 delta network with two paths from specific inputs to outputs indicated.

Solution 4.23.

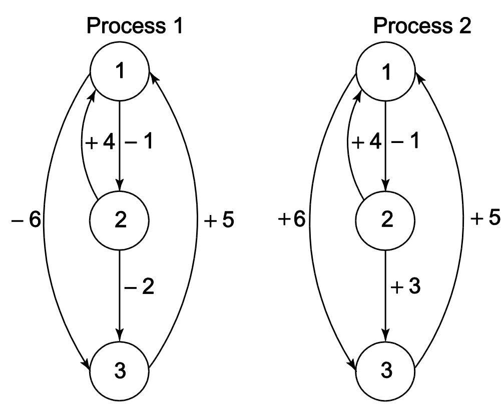
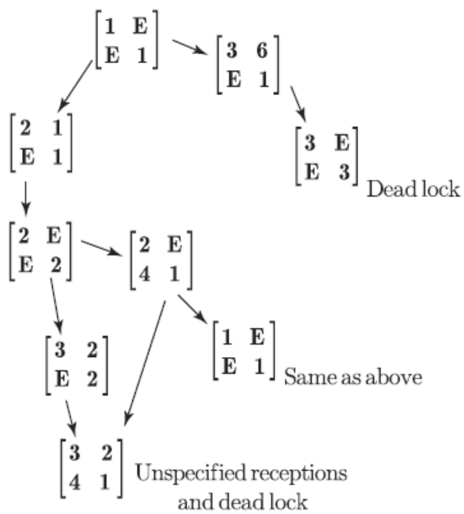
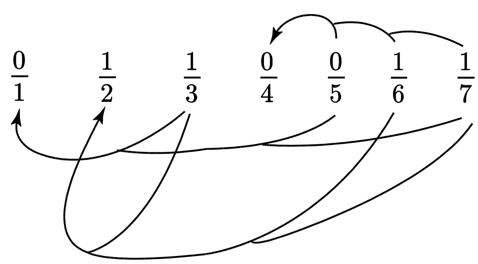


Fig. 4.17. A reachability tree problem

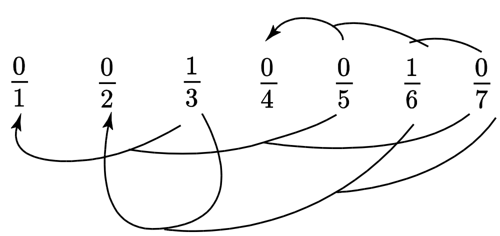


Solution 4.24. 1011000 $\rightarrow$ odd parity so there is an error.

Solution 4.25.



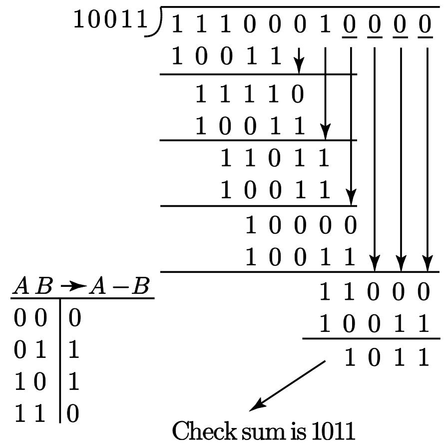
Solution 4.26.



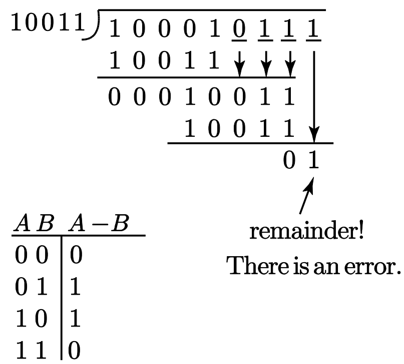
1: odd $\searrow$
2: even $1 + 4 = 5$, 5th bit in error
4: odd $\nearrow$

1: odd $\searrow$
2: even $1+4=5$, 5th bit in error 4: odd
2: odd $\nearrow$

Solution 4.27.



Solution 4.28.



Solution 4.29.

4B5B	01001	10100	11110
Binary	0001	0010	0000
Decimal	1	2	0

Solution 4.30. Networking Coding:

b1	1	0	1	0	0	1	0	1	1	0
b1+b2	0	1	1	0	0	1	0	0	0	1
Final b2	1	1	0	0	0	0	0	1	1	1

b1	b2	b1+b2
0	0	0
0	1	1
1	0	1
1	1	0

Solution 4.31.

Basis	0	1

+	↑	→
×	↗	↖

Alice’s Bits	1	1	0	0	1	1	1	0
Alice’s Random Basis	+	×	×	+	+	×	×	×
Alice’s Xmssn	→	↖	↗	↑	→	↖	↖	↗
Bob’s Basis	+	×	×	×	+	+	×	+
Bob’s Measurements	→	↖	↗	↖	→	→	↖	↑
Secret Shared Key	1	1	0	_	1	_	1	_

5

Divisible Load Modeling for Grids

1. A divisible load is a communication/computing load that can be arbitrarily partitioned among processors and links. There is thus a very fine granularity to the load (there have been some papers on coarser granularity). Continuous variables are used to represent the fraction of load assigned to a given link or processor. There are also no precedence relations among the data.
2. The basic divisible load model is a linear model which leads to the usual tractable mathematics of linear models such as those for queueing models or electric circuits.
3. Consider a single node distributing load to its children nodes in a single level tree (star) type network. Under sequential distribution load is distributed to one child at a time. Under simultaneous distribution load is distributed to all children concurrently. A hybrid load distribution strategy is a multi-installment (multi-round) distribution where load is sent sequentially to children from the root in repeating installments (rounds).
4. As the number of children increases when using sequential load distribution, a saturation in performance (solution time and speedup) is observed. This is due to the fact that no matter

how many children there are, at any point in time only one child is receiving load. Simultaneous scheduling avoids this problem by feeding load concurrently to all children present.

5. Under staggered start, each child processor in a single level tree network first receives load from the root and only after all of this load is received starts computing. Under simultaneous start a child node starts computing as soon as load begins to be received and reception of load and computing may go on concurrently for some period of time.
6. A front end processor performs communication duties for a main processor, thus the main processor can concentrate on computation. The use of front ends leads to a faster solution as communication and computation can proceed concurrently.
7. Speedup is defined as the ratio of the time to solve a computational problem on one processor divided by the time for N homogeneous processors to solve the problem. Since the N processor solution should be faster than the one processor solution the ratio is greater than 1.0. However speedup is usually less than N because of overhead, communication delays, and serialization.
8. In the Gantt type chart there is one graph per processor. Time is on the horizontal axis. Communication is shown above the horizontal axis and computation is shown below it. Usually the chart starts at time $t = 0$.
9. Speedup increases nonlinearly for nonlinear models because of the nonlinear nature of the computing time. It is more efficient to process load in fragments on a number of processors than on one processor. However the results of individual processors processing fragments of load must often be combined in a “post-processing” phase of computation that adds overhead. The amount of improvement in overall speedup for nonlinear problems is thus problem specific.
10. An equivalent processor is a processor that has identical operating characteristics (solution time and speedup) as the network of processors and links that it replaces. The concept of equivalent elements is a basic property of linear models. To find the overall performance of a network of elements one can aggregate elements until a single processor results with known processing speed that is identical to the processing speed of the overall network.
11. As the number of nodes is increased in a linear daisy chain under store and forward load distribution, load for the further processors has to be relayed from the load originating

processor over many links to where it is actually processed. This repetitive transmission of the same load is a form of overhead that results in a saturated network performance. The same effect occurs for store and forward load distribution in tree networks as depth is increased. The use of virtual cut through switching, where load is relayed to the next node before being completely received, can mitigate this saturation.

12. It is not worth distributing load from one node to the next in a linear daisy chain if the link speed is too slow relative to the computing speed of the first node. In this case one obtains a faster solution if the first node processes the remaining load rather than communicating it to the next node.
13. Many (often unpredictable) events occur in a computer that take effort on the part of the processor, thus leaving less available processor effort for a divisible job(s) of interest. Such events include background job(s) start and termination.
14. An indivisible load is a load or task that must be integrally processed on a single processor. It cannot be divided up among multiple processors in a divisible sense. However often in the indivisible load literature it is assumed that multiple indivisible tasks can be assigned to a single processor.
15. Many software programs are not divisible.
16. Divisible load scheduling problems can often be solved by either linear equation solution or linear programming. It depends on the number of equations and variables.

Solution 5.17.

$$\text{speedup} = 1 + k_1\left[1 + \sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]$$

$$q_l = 1-\sigma \quad \text{for homogeneous new where} \quad \sigma = zT_{cm} / wT_{cp}$$

$$\text{speedup} = 1 + k_1\left[1 + \sum_{i=2}^{m}(1-\sigma)^{i-1}\right] \quad \text{Let } i \to i+1$$

$$\text{speedup} = 1 + k_1\left[1 + \sum_{i=1}^{m-1}(1-\sigma)^{i}\right]$$

$$= 1 + k_1\left[\sum_{i=0}^{m-1}(1-\sigma)^{i}\right]$$

$$= 1 + k_1\left[\frac{1-(1-\sigma)^m}{\sigma}\right] \quad \text{as} \quad \sum_{n=0}^{N} X^n = \frac{1-X^{N+1}}{1-X} \quad \text{and}$$

$$k_1 = w_0 / w_1$$

so

$$\text{speedup} = 1 + \frac{w_0}{w_1}\left[\frac{1-(1-\sigma)^m}{\sigma}\right] \tag{5.21}$$

Solution 5.18. No, it is just the nature of nonlinear mathematics. As a non-scheduling example, for wireless communications many short hops use less energy than a single big hop because of the nonlinear dependency of energy on transmission distance.

Solution 5.19. An equivalent element, in a network with linear properties, is a single element that has identical operating characteristics (processing speed for divisible load theory or resistance in a linear electric network) as the original network it replaces. Sometimes equivalent elements are used as part of an analysis and sometimes to simplify the physical network.

Solution 5.20. Product form solutions allow simplified and often computationally efficient solutions.

Solution 5.21. Many networks have a saturating performance. Adding additional hardware at some point does not lead to much improvement. So if you know a 10 processor system has 98% of the performance of an infinite number of processors system, you have justification to use the 10 processor system.

Solution 5.22. If you are renting computer time to users on an expensive computational facility, monetary cost optimization is important.

Solution 5.23. In searching for extraterrestrial signals, an artificial signal coming from space is a signature.

Solution 5.24.

T_{cm} : Time to transmit entire load over channel.

T_{cp} : Time to process entire load on a processor.

(These definitions are somewhat different from those in the rest of the chapter).

(a) Improperly formulated—my apologies.

(b) (Cheng 88) Since T_{cp} / N is overlapped with T_{cm} / N except for the last processor,

$$
\begin{aligned}
T_T &= \frac{T_{cm}}{N}(N-1+N-2+\cdots+1)+\frac{T_{cp}}{N} \\
&= \frac{N-1}{2}T_{cm}+\frac{T_{cp}}{N}
\end{aligned}
$$

$$
\frac{dT_T}{dN}=0 \Rightarrow N_{\text{optima1}}=\sqrt{\frac{2T_{cp}}{T_{cm}}}=\sqrt{2/\rho}
$$

(c)

$$
\begin{aligned}
T_T &= \frac{T_{cm}}{N}(N-1+N-2+\cdots+1)+\frac{T_{cp}}{N} \\
&\quad +T_s(1+2+3+\cdots+N-1) \\
T_T &= \frac{N-1}{2}T_{cm}+\frac{T_{cp}}{N}+T_s\frac{(N-1)N}{2}
\end{aligned}
$$

$$
\frac{dT_T}{dN}=0 \Rightarrow N^3T_s+N^2T_{cm}-2T_{cp}=0
$$

and solve for N_{optima1}.

Solution 5.25. (a) *Sequential*:

$$
q_i=1-\sigma=1-\frac{zT_{cm}}{wT_{cp}}=1-\frac{1}{3}=\frac{2}{3} \tag{5.20}
$$

$$
w_{eq}=\frac{1}{1+k_1\left[1+\sum_{i=2}^{m}\left(\prod_{l=z}^{1}q_l\right)\right]}w_0 \tag{5.16}
$$

$$
\text{but}\quad k_1=\frac{w_0}{w_1}=1
$$

$$
\text{so}\quad w_{eq}=\frac{6}{1+1\left[1+\frac{1}{3}+\left(\frac{1}{3}\right)^2\right]}=2.45
$$

$$
\text{speedup}=1+\frac{w_0}{w}\left[\frac{1-(1-\sigma)^m}{\sigma}\right] \tag{5.21}
$$

$$
=1+\left[\frac{1-(2/3)^3}{1/3}\right]=3.11
$$

For the α's:

$$\alpha_1 = \frac{1}{\frac{1}{k_1} + 1 + \sum_{i=2}^{m}\left(\prod_{l=2}^{1} q_l\right)} = \frac{1}{1+1+\frac{2}{3}+\left(\frac{2}{3}\right)^2} = 0.3214$$

$$\alpha_0 = \frac{1}{k_1}\alpha_1 = \alpha_1 = 0.3214$$

$$\alpha_2 = q_i\,\alpha_1 = \frac{2}{3}0.3214 = 0.2143 \quad \sum \alpha_i \approx 1$$

$$\alpha_3 = q_i\,\alpha_2 = \frac{2}{3}0.2143 = 1.428$$

(b) *Simultaneous Distribution with Staggered Start*

(5.31)

$$w_{eq} = \frac{1}{k_1}\alpha_1\, w_0 = \frac{4}{3} \times 0.2308 \times 6 = 1.84 \quad \text{where } T_{cp} = 1$$

$$\text{speedup} = 1 + k_1 m = 1 + \frac{3}{4} \times 3 = 3.25$$

From above, for α's, $\alpha_1 = 0.2308$

$$\alpha_0 = \frac{1}{k_1}\alpha_1 = \frac{4}{3} \times 0.2308 = 0.3077$$

$$\alpha_2 = q\,\alpha_1 = 1 \times 0.2308 = 0.2308$$

$$\alpha_3 = q\,\alpha_2 = 1 \times 0.2308 = 0.2308 \quad \sum \alpha_i \cong 1$$

(c) *Simultaneous Distribution with Simultaneous Start*

$$\alpha_1 = \frac{1}{\left[\frac{1}{k_1} + 1 + \sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]} \tag{5.51}$$

$$k_1 = \frac{w_0}{w_1} = 1 \quad q_l = 1$$

$$\alpha_1 = \frac{1}{\left[1+1+1+1^2\right]} = 0.25$$

$$w_{eq} = \frac{1}{k_1}\alpha_1 w_0 = 1 \times 0.25 \times 6 = 1.5 \quad \text{where } T_{cp} = 1$$

$$\text{speedup} = 1 + k_1 m = 4.0$$

For the α 's:

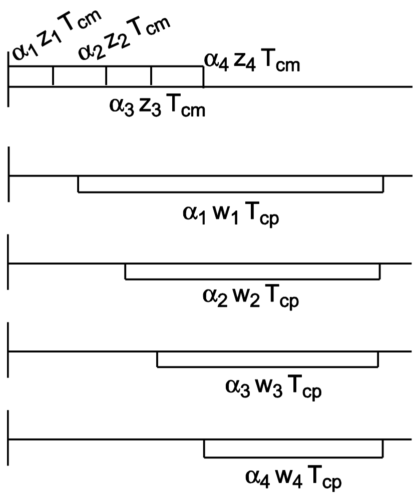
$$\alpha_0 = \frac{1}{k_1}\alpha_1 = \alpha_1 = 0.25$$
$$\alpha_2 = q\,\alpha_1 = \alpha_1 = 0.25$$
$$\alpha_3 = q\,\alpha_2 = \alpha_2 = 0.25 \qquad \sum \alpha_i\text{'s} = 1$$

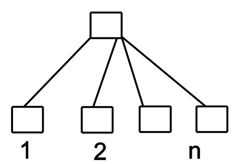
(d) The w_{eq} and speedup are decreasing (increasing), as they should be, since sequential is the worst strategy, and simultaneous distribution with simultaneous start is the best strategy.

	Sequential	Simul. dist., staggered start	Simul. dist., simul. start
w_{eq}	2.45	1.84	1.5
Speedup	3.11	3.25	4.0

Note that all speedups ≤ 4 (there are four processors).

Solution 5.26.





(a) Timing equations:

$$N-1eqs\begin{cases}\alpha_1 w_1 T_{cp} = \alpha_2 z_2 T_{cm} + \alpha_2 w_2 T_{cp}\\ \alpha_2 w_2 T_{cp} = \alpha_3 z_3 T_{cm} + \alpha_3 w_3 T_{cp}\\ \vdots\\ \alpha_i w_i T_{cp} = \alpha_{i+1} z_{i+1} T_{cm} + \alpha_{i+1} w_{i+1} T_{cp}\\ \vdots\\ \alpha_{N-1} w_{N-1} T_{cp} = \alpha_N z_N T_{cm} + \alpha_N w_N T_{cp}\end{cases}$$

$$+\alpha_1 \alpha_2 + \cdots + \alpha_N = 1$$

(b)

$$\alpha_i = \frac{w_{i-1}T_{cp}}{z_iT_{cm} + w_iT_{cp}}\alpha_{i-1}$$

$$= q_i\alpha_{i-1} = \left(\prod_{l=2}^{i} q_l\right)\times\alpha_1$$

$$\alpha_1 + \alpha_2 + \cdots + \alpha_N = 1$$

$$\left[1+\sum_{i=2}^{N}\left(\prod_{l=2}^{i} q_l\right)\right]\alpha_i = 1$$

$$\alpha_1 = \frac{1}{1+\sum_{i=2}^{N}\left(\prod_{l=2}^{i} q_l\right)}$$

(c)

$$T_f = \alpha_1(z_1T_{cm} + w_1T_{cp}) = \frac{z_1T_{cm} + w_1T_{cp}}{1+\sum_{i=2}^{N}\left(\prod_{l=2}^{i} q_l\right)}$$

$$\text{speedup} = \frac{T_0}{T_f} = \frac{wT_{cp}\left(1+\sum_{i=2}^{N}\left(\prod_{l=2}^{i} q_l\right)\right)}{zT_{cm} + wT_{cp}}$$

$$= \frac{wT_{cp}}{zT_{cm} + wT_{cp}}\left(1+\sum_{i=2}^{N}\left(\frac{wT_{cp}}{zT_{cm} + wT_{cp}}\right)^{i-1}\right)$$

can be further simplified using summation formula.

Solution 5.27. (Cheng 88, Appendix)

Case I: The N^{th} processor stops first. Assume $w = z = 1$ without loss of generality.

$$T_T = \alpha T_{cp}$$

$$\alpha T_{cp} \geq (1-\alpha)(T_{cm} + T_{cp})$$

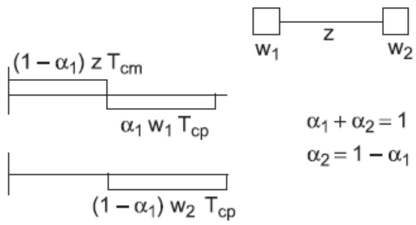
$$\alpha \geq \frac{T_{cm} + T_{cp}}{T_{cm} + 2T_{cp}} \qquad (*)$$

$$\min\ (T_T) = \min\ (\alpha)T_{cp}$$

where $\min(\alpha)$ occurs when $(*)$ has equality or both processors stop at the same time.

Case II: Similar to Case I.

Solution 5.28.



$$
\begin{aligned}
\alpha_1 w_1 T_{cp} &= (1-\alpha_1) w_2 T_{cp} \\
\alpha_1 (w_1 T_{cp} + w_2 T_{cp}) &= w_2 T_{cp} \\
\alpha_1 &= \frac{w_2 T_{cp}}{w_1 T_{cp} + w_2 T_{cp}} = \frac{w_2}{w_1 + w_2} \\
T_f &= (1-\alpha_1) z T_{cm} + \alpha_1 w_1 T_{cp} \\
&= z T_{cm} + \alpha_1 (w_1 T_{cp} - z T_{cm})
\end{aligned}
$$

$w_1 T_{cp}$ should be $> z T_{cm}$, otherwise seek to $\max \alpha_1$ and all load processed on P_1 as communication is too slow.

Solution 5.29.

(a)

$$\text{speedup}\,|_{\text{staggered start}} = 1 + k_1\, m = 1 + \frac{w_0 T_{cp}}{w T_{cp} + z T_{cm}}\, m$$

$$\text{If } \; z T_{cm} \to 0$$

$$\text{speedup}\big|_{\text{staggered start}} = 1 + \frac{w_0}{w}\, m = \text{speedup}\big|_{\text{simultaneous start}}$$

(b)

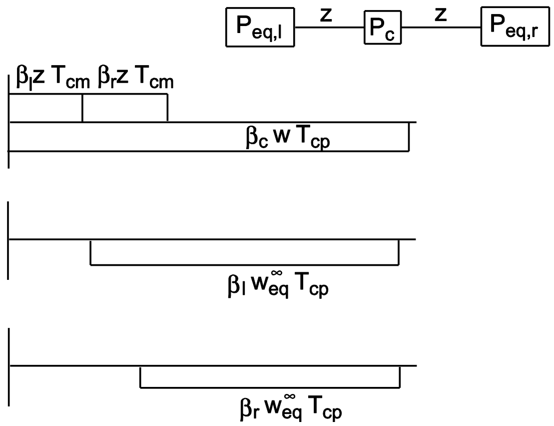
$$1 + \frac{w_0}{w}\, m \overset{?}{>} 1 + \frac{w T_{cp}}{w T_{cp} + z T_{cm}}\, m$$

$$\frac{\cancel{w_0}}{w} > \frac{\cancel{w_0}\, T_{cp}}{w T_{cp} + z T_{cm}}$$

Cross-multiply:

$$w T_{cp} + z T_{cm} > w T_{cp} \quad \text{confirmed!}$$

Solution 5.30.



(a)

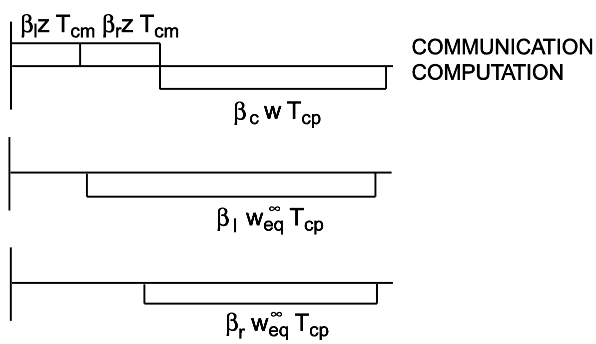
$$
\begin{aligned}
T_{fe}^{\infty} &= \beta_c w T_{cp} \\
&= \beta_l z T_{cm} + \beta_l w_{eq}^{\infty} T_{cp} \\
&= (\beta_l + \beta_r) z T_{cm} + \beta_r w_{eq}^{\infty} T_{cp} \\
&= \underset{\text{system}}{\overset{w_{eq,s},\, T_{cp}}{\uparrow}} \qquad \beta_c + \beta_r + \beta_l = 1
\end{aligned}
$$

(b) Using algebra:

$$
w_{eq,s} = \frac{w(zT_{cm} + w_{eq}^{\infty} T_{cp})}{zT_{cm} + w_{eq}^{\infty} T_{cp} + wT_{cp} + \dfrac{w w_{eq}^{\infty} T_{cp}^2}{zT_{cm} + w_{eq}^{\infty} T_{cp}}}
$$

(c) Calculate w_{eq}^{∞} using boundary results and then substituting in (b).

Solution 5.31. (Bataineh 97)

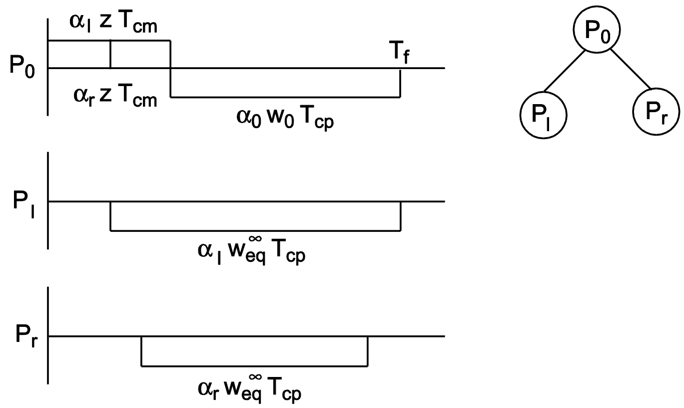


(a)

$$
\begin{aligned}
T_{nfe}^{\infty} &= (\beta_l + \beta_r) z T_{cm} + \beta_c w T_{cp} \\
&= \beta_l z T_{cm} + \beta_l w_{eq}^{\infty} T_{cp} \\
&= (\beta_l + \beta_r) z T_{cm} + \beta_r w_{eq}^{\infty} T_{cp} \\
&= w_{eq,s}^{\infty} T_{cp} \qquad \beta_c + \beta_l + \beta_r = 1
\end{aligned}
$$

(b)
$$
w_{eq,s}^{\infty} = \frac{w(zT_{cm} + w_{eq}^{\infty} T_{cp})^2}{(w_{eq}^{\infty} T_{cp})^2 + w_{eq}^{\infty} T_{cp}^2 w + w T_{cp} (z T_{cm} + w_{eq}^{\infty} T_{cp})}
$$

Solution 5.32.



(a) Timing equations:

$$
\begin{aligned}
T_f &= (\alpha_l + \alpha_r) z T_{cm} + \alpha_0 w T_{cp} \\
&= \alpha_l z T_{cm} + \alpha_l w_{eq}^{\infty} T_{cp} \\
&= (\alpha_l + \alpha_r) z T_{cm} + \alpha_r w_{eq}^{\infty} T_{cp} = w_{eq,s}^{\infty} T_{cp}
\end{aligned}
$$

(b) With algebra:

$$
w_{eq,s}^{\infty} = \frac{w(zT_{cm} + w_{eq}^{\infty} T_{cp})^2}{(w_{eq}^{\infty} T_{cp})^2 + 2 w w_{eq}^{\infty} T_{cp}^2 + w z T_{cp} T_{cm}}
$$

(c) With more algebra:

$$
(w_{eq}^{\infty})^3 + w(w_{eq}^{\infty})^2 - [w\, z\, \rho] w_{eq}^{\infty} - w\, z^2 \rho^2 = 0
$$

Solution 5.33.

(a) (Sohn):

$$
T_f - T_n = \alpha_n w_n T_{cp} \;\; n = 1, 2, \ldots, N
$$

$$
T_n - T_{n-1} = \alpha_n \overset{-n}{z}_{n-1}(T) T_{cm} n = 1, 2, \ldots, N
$$

$$\bar{z}_{n-1}^{n}(T) = \left(E\left\{ \frac{1}{z_{n-1}^{n}(T)} \right\} \right)^{-1}$$

$$= \frac{T_n - T_{n-1}}{\int_{T_{n-1}}^{T_n} \frac{1}{z(T)} dT}$$

$z_{n-1}^{n}(T)$ is inverse of the time avg. of the applied channel speed in interval $(T_{n\text{-}1}, T_n)$

$$\frac{d}{dT} z(T) = \sum_{k=0}^{\infty} S_k \delta(t - t_k) z$$

$$S_k = \begin{cases} +1 & \text{for arrival} \\ -1 & \text{for departure} \end{cases}$$

$$z(T) = \sum_{k=0}^{\infty} S_k \mu(t - t_k) z$$

$$\frac{1}{z(T)} = \sum_{k=0}^{\infty} \left(\sum_{j=0}^{k} S_j \right)^{-1} [\mu(t - t_k) - \mu(t - t_{k+1})] \frac{1}{z}$$

$$\int_{T_{n-1}}^{T_n} \frac{1}{z(T)} = \frac{T_n}{z(T_n)} - \frac{T_{n-1}}{z(T_{n-1})}$$

$$T_n - T_{n-1} = \alpha_n \bar{z}_{n-1}^{n}(T) T_{cm}$$

$$= \alpha_n T_{cm} \frac{T_n - T_{n-1}}{\int_{T_{n-1}}^{T_n} \frac{1}{z(t)} dt}$$

$$\alpha_n = \frac{1}{T_{cm}} \int_{T_{n-1}}^{T_n} \frac{1}{z(T)} dT$$

$$= \frac{1}{T_{cm}} \left[\frac{T_n}{z(T_n)} - \frac{T_{n-1}}{z(T_{n-1})} - \sum_{k=x_{n-1}+1}^{x_n} \left(\frac{1}{z(t_k)} - \frac{1}{z(t_{k-1})} \right) t_k \right]$$

(b) One can solve for $\alpha's$ versus finish time and select those $\alpha's$ that sum to one as being correct (Sohn 98).

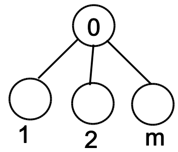
Solution 5.34.

$$\alpha_n = \frac{1}{T_{cp}} \int_{T_n}^{T_f} \frac{1}{w_n(T)} dt$$

$$\sum_{i=1}^{n} \alpha_i = \frac{1}{T_{cm}} \int_{0}^{T_n} \frac{1}{z(T)} dt$$

see (Sohn 98) for full explanation.

Solution 5.35.

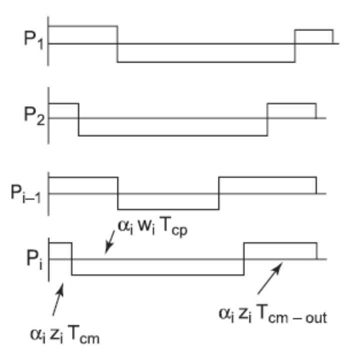


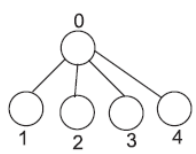
$$\text{Start with}\begin{cases} \alpha_1 w_1 T_{cp} \le T_f \\ \alpha_1 z_1 T_{cm} + \alpha_2 w_2 T_{cp} \le T_f \\ \alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_3 w_3 T_{cp} \le T_f \\ \left(\sum_{i=1}^{m} \alpha_i z_i T_{cm} + \alpha_m w_m \tau_{cp}\right) \le T_f \\ T_f = \alpha_0 w_0 T_{cp} \end{cases}$$

m.p.: $\min T_f = \min \alpha_0 w_0 T_{cp}$

$$\begin{aligned} \alpha_1 w_1 T_{cp} - \alpha_0 w_0 T_{cp} &\le 0 \\ \alpha_1 z_1 T_{cm} + \alpha_2 w_2 T_{cp} - \alpha_0 w_0 T_{cp} &\le 0 \\ \alpha_1 z_1 T_{cm} + \alpha_2 z_2 T_{cm} + \alpha_3 w_3 T_{cp} - \alpha_0 w_0 T_{cp} &\le 0 \\ \sum_{i=1}^{m} \alpha_i z_i T_{cm} + \alpha_n w_m T_{cp} - \alpha_0 w_0 T_{cp} &\le 0 \\ \alpha_0 + \alpha_1 + \alpha_2 + \cdots + \alpha_m - 1 &= 0 \\ \alpha_1, \alpha_2, \ldots, \alpha_m &\ge 0 \end{aligned}$$

Solution 5.36.





$$\alpha_1(z_1 T_{cm} + w_1 T_{cp} + z_1 T_{cm-out}) = \alpha_2(z_2 T_{cm} + w_2 T_{cp} + z_2 T_{cm-out})$$

Let

$$q_i = \frac{z_{i-1}T_{cm} + w_{i-1}T_{cp} + z_{i-1}T_{cm-out}}{z_iT_{cm} + w_iT_{cp} + z_iT_{cm-out}}$$

$$\text{Then} \quad \alpha_i = q_i\alpha_{i-1} \quad i = 2,3,\ldots,m$$

Then we can use the equations of Sect. 5.2.2 with the new q_i to solve for the $\alpha's$ and the speedup.

Solution 5.37. *(From Hung PhD Thesis, 2003)*:

Each child begins to receive load once the root has received the load for itself and all its children. The root without data storage scheduling in Fig. 5.5 is of interest as it represents a typical store and forward switching behavior where the entire load for a subtree is received by the parent (root) before the children commence receiving and processing load. Each processor begins computing the received data as soon as the data starts to arrive. Other scheduling variations are certainly possible.

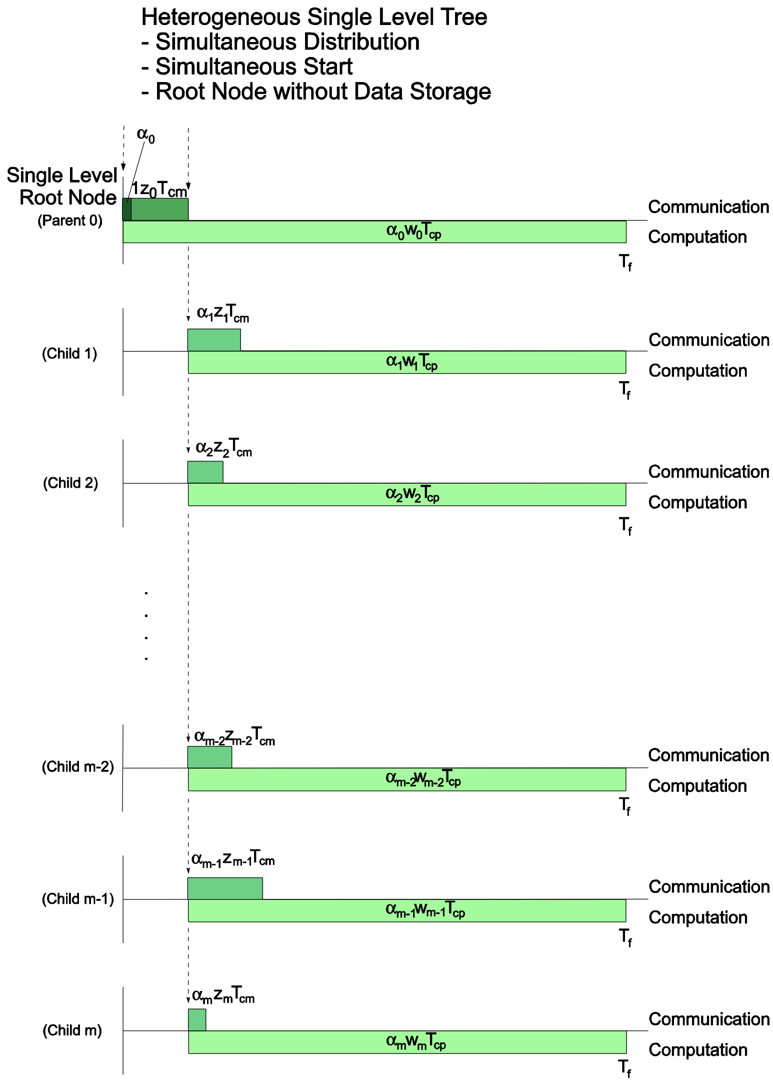


Fig. 5.5. Timing diagram of single level tree using simultaneous distribution and simultaneous start for a root node without data storage

From the timing diagram of Fig. 5.5, the fundamental recursive equations of the system can be formulated as follows:

$$\alpha_0 w_0 T_{cp} = 1 \cdot z_0 T_{cm} + \alpha_1 w_1 T_{cp} \tag{5.24}$$

$$\alpha_{i-1} w_{i-1} T_{cp} = \alpha_i w_i T_{cp} \quad i = 2, 3, \ldots, m \tag{5.25}$$

The normalization equation for this single level tree is

$$\alpha_0 + \alpha_1 + \alpha_2 + \cdots + \alpha_m = 1 \tag{5.26}$$

This gives $m+1$ linear equations with $m+1$ unknowns.

From (5.24), we obtain

$$\alpha_0 = \frac{z_0 T_{cm}}{w_0 T_{cp}} + \frac{w_1 T_{cp}}{w_0 T_{cp}} \alpha_1 = \sigma_0 + \frac{1}{k_1} \alpha_1 \tag{5.27}$$

Here σ_0 is defined as $(z_0 T_{cm})/(w_0 T_{cp})$ and k_1 as w_0 / w_1. From (5.25), the solution of α_i is

$$\alpha_i = \frac{w_{i-1} T_{cp}}{w_i T_{cp}} \alpha_{i-1} = q_i \alpha_{i-1} \quad i = 2, 3, \ldots, m \tag{5.28}$$

where $q_i = w_{i-1} / w_i$.

Equation (5.28) can be represented as

$$\alpha_i = q_i \alpha_{i-1} = \left(\prod_{l=2}^{i} q_l \right) \alpha_1 \; i = 2, 3, \ldots, m \tag{5.29}$$

Employing (5.27) and (5.29), the normalization equation (5.26) becomes

$$\sigma_0 + \frac{1}{k_1} \alpha_1 + \alpha_1 + \sum_{2}^{m} \alpha_i = 1 \tag{5.30}$$

$$\left[\frac{1}{k_1} + 1 + \sum_{i=2}^{m} \left(\prod_{l=2}^{i} q_l \right) \right] \alpha_1 = 1 - \sigma_0 \tag{5.31}$$

Consequently,

$$\alpha_1 = \frac{1 - \sigma_0}{\left[\frac{1}{k_1} + 1 + \sum_{i=2}^{m} \left(\prod_{l=2}^{i} q_l \right) \right]} \tag{5.32}$$

From Fig. 5.5, the finish time is achieved as:

$$T_{f,m} = \alpha_0 w_0 T_{cp} = \left[\sigma_0 + \frac{1}{k_1} \alpha_1 \right] w_0 T_{cp} \tag{5.33}$$

The term, $T_{f,m}$, is the finish time of a single divisible job solved on the entire tree, consisting of one root node as well as m child nodes. Now, collapsing a single level tree into a single node, one can obtain the finish time of the single level tree as follows.

$$T_{f,m} = w_{eq} T_{cp} = \alpha_0 w_0 T_{cp} = \left[\sigma_0 + \frac{1}{k_1} \alpha_1 \right] w_0 T_{cp} \tag{5.34}$$

On the other hand, $T_{f,0}$ is defined as the solution time for the entire divisible load solved on the root processor.

$$T_{f^0}, = \alpha_0 w_0 T_{cp} = 1 \times w_0 T_{cp} = w_0 T_{cp} \tag{5.35}$$

According to Definition 1 in Sect. 5.2, $\gamma_{eq} = w_{eq} / w_0 = T_{f,m} / T_{f,0}$, one obtains the value of γ_{eq} by (5.35) dividing (5.34[SD1]).

$$\begin{aligned}\gamma_{eq} &= \sigma_0 + \frac{1}{k_1}\alpha_1 \\ &= \sigma_0 + \frac{1}{k_1} \times \frac{1-\sigma_0}{\left[\frac{1}{k_1} + 1 + \sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]} \\ &= \frac{1+\sigma_0 k_1\left[1+\sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]}{1+k_1\left[1+\sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]}\end{aligned} \tag{5.36}$$

Speedup is intuitively obtained by $T_{f,0} / T_{f,m}$, or $1/\gamma_{eq}$; therefore,

$$\text{Speedup} = \frac{1}{\gamma_{eq}} = \frac{1+k_1\left[1+\sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]}{1+\sigma_0 k_1\left[1+\sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]} \tag{5.37}$$

Two cases are discussed:

1) *General Case*: Since $\prod_{l=2}^{i} ql$ can be simplified as w_1 / w_i, γ_{eq} can be derived from (36) as

$$\gamma_{eq} = \frac{1+\sigma_0 \frac{w_0}{w_1}\left[1+\sum_{i=2}^{m}\frac{w_1}{w_i}\right]}{1+\frac{w_0}{w_1}\left[1+\sum_{i=2}^{m}\frac{w_1}{w_i}\right]} = \frac{1+\sigma_0 w_0 \sum_{i=1}^{m}\frac{1}{w_i}}{1+w_0\sum_{i=1}^{m}\frac{1}{w_i}} \tag{5.38}$$

Thus, the value of speedup becomes

$$\text{Speedup} = \frac{1+w_0\sum_{i=1}^{m}\frac{1}{w_i}}{1+\sigma_0 w_0\sum_{i=1}^{m}\frac{1}{w_i}} \tag{5.39}$$

2) *Homogeneous Case*: As a special case, consider the situation of a homogeneous network where all children processors have the same inverse computing speed and all links have the same inverse transmission speed. In other words, $w_i = w$ and $z_i = z$ for $i = 1, 2, \ldots, m$. Note that the root inverse computing speed, w_0 can be different from those $w_i, i = 1, 2, \ldots, m$. Consequently,

$$k_1 = \frac{w_0}{w_1} = \frac{w_0}{w} \tag{5.40}$$

$$q_i = \frac{w_{i-1}}{w_i} = \frac{w}{w} = 1 \quad i = 2, 3, \ldots, m$$

$$\gamma_{eq} = \frac{1+\sigma_0 k_1\left[1+\sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]}{1+k_1\left[1+\sum_{i=2}^{m}\left(\prod_{l=2}^{i} q_l\right)\right]} \tag{5.41}$$

$$= \frac{1+\sigma_0 m \frac{w_0}{w}}{1+m\frac{w_0}{w}} = \frac{w+\sigma_0 m w_0}{w+m w_0}$$

$$\text{Speedup} = \frac{1+m\frac{w_0}{w}}{1+\sigma_0 m\frac{w_0}{w}} = \frac{w+m w_0}{w+\sigma_0 m w_0} \tag{5.42}$$

Solution 5.38.

$$\sum_{i=1}^{N}\alpha_{i,j} w_i T_{cp} = \sum_{i=1}^{N}\alpha_{i,j+1} w_{j+1} T_{cp} \quad j = 1, 2, \ldots, M-1$$

$$\sum_{i=1}^{N}\alpha_j L_i w_j = \sum_{i=1}^{N}\alpha_{j+1} L_i w_{j+1} \quad j = 1, 2, \ldots, M-1$$

Using this and $\sum_{i=1}^{M}\alpha_i = 1$,

$$\alpha_j = \frac{1}{w_j\left(\sum_{x=1}^{M}\frac{1}{w_x}\right)} \quad j = 1, 2, \ldots, M$$

$$\alpha_{i,j} = \frac{1}{w_j\left(\sum_{x=1}^{M}\frac{1}{w_x}\right)} L_i$$

$$T(M) = \sum_{i=1}^{N}\alpha_{i,M} w_M T_{cp}$$

6

Problem 1)

$$P=10$$

$$S_{Amdahl}=\frac{1}{(1-f)-1\frac{F}{P}}\quad \begin{array}{l} F=0.9\ \ \frac{1}{0.1+0.09}=\frac{1}{0.19}=5.3 \\ F=0.95\ \ \frac{1}{0.05+0.95}=\frac{1}{0.145}=6.9 \\ F=0.98\ \ \frac{1}{0.02+0.98}=\frac{1}{0.118}=8.47 \end{array}$$

$$P=10$$

$$S_{Gustafson}=(1-f)+\gamma f\Big|\begin{array}{l} F=0.9\quad 0.1+9=9.1 \\ F=0.95\quad 0.5+9.5=9.55 \\ F=0.98\quad 0.7+9.8=9.82 \end{array}$$

$$P=10\qquad \sqrt{P}=3.16$$

$$S^{General}=\frac{(1-f)+F\sqrt{P}}{(1-f)+\frac{F\sqrt{P}}{P}}\qquad F=0.9\,\frac{0.1+0.9\times 3.16-2.944}{0.1+{}^{0.9}\!/_{3.16}-0.385}=7.65$$

$$F=0.95\,\frac{0.05+0.95\times 3.16}{0.05+0.95/3.16}=\frac{3.052}{0.351}=8.67$$

$$\frac{\sqrt{P}}{P}=\frac{1}{\sqrt{P}}\qquad F=0.98\,\frac{0.02+0.98\times 3.16}{0.02+{}^{0.98}\!/_{3.16}}=\frac{3.117}{0.33}=9.44$$

Note for given f[SD2]

$S_{Amdahl} < S_{General} < S_{Gustafson}$

2)

$$F=0.9$$

$$S_{Amdahl}=\frac{1}{(1-f)+\frac{F}{P}}\qquad P=5\,\frac{1}{0.1+\frac{0.9}{5}}=\frac{1}{0.28}=3.57$$

$$P=10\,\frac{1}{0.1+\frac{0.9}{10}}=\frac{1}{0.19}=5.26$$

$$P=15\,\frac{1}{0.1+\frac{0.9}{15}}=\frac{1}{0.16}=6.25$$

$$S_{Gustafson}=(1-f)+Fp\quad \begin{array}{l} P=5\quad 0.1+(0.9)(5)=4.6 \\ P=10\quad 0.1+(0.9)(10)=9.1 \\ P=15\quad 0.1+(0.9)(15)=13.6 \end{array}$$

$$S_{General} = \frac{(1-f) + F\sqrt{P}}{(1-f) + \frac{F\sqrt{P}}{P}}$$

$$P = 5 \quad \frac{0.1 + (0.9)(2.24)}{0.1 + 0.9/2.24} = \frac{2.12}{0.502} = 4.22$$

$$P = 10 \quad \frac{0.1 + (0.9)(3.16)}{0.1 + 0.9/3.16} = \frac{2.94}{0.385} = 7.64$$

$$P = 15 \quad \frac{0.1 + (0.9)(3.88)}{0.1 + 0.9/3.88} = \frac{3.59}{0.332} = 10.8$$

$S_{Amdahl} < S_{General} < S_{Gustafson}$

Solutions

6.1 and 6.2 see separate numerical solutions.

6.3 The assumption is that are p processors that reduce the parallel execution time of a job by a simple factor of p (if there were p = 4 processors and parallel execution time is 100 ms, applying the four processors to the problem results in an execution time of 25 ms[SD3]).

6.4 Amdahl's law gives an idea as to what extent a component improvement can affect the entire system performance.

6.5 Amdahl's law assumes a constant problem size no matter how many processors are applied to a problem. Gustafson felt that as technology and time advance, more cores are used to solve larger/more complicated problems. Thus one could have a parallel fraction of load that grows linearly in problem size (that is using fp instead of f).

6.6 A higher speedup is better than a lower speedup—if one has 30 processors one would like an effective processing power as close to 30 as possible.

6.7 The speedup expressions of different multicore architectures are useful for providing guidance in answering practical questions such as the right mix of small and large cores in a system.

6.8 It is important to consider metrics such as average power consumption and performance per average power because power is an important constraint in limiting the size and processing power of systems that can be configured.

6.9 Under sequential processing in Sect. 6.8 sequential processing, CPU and GPU parts of the processing occur at separate times. In some cases the CPU and GPU parts can proceed in parallel (concurrently).

6.10 In general objective functions can be minimized or maximized to achieve optimal performance for a given model and parameters.

6.11 The basic question that Sect. 6.10 addresses is whether it is better to process data locally (on a mobile device) or remotely (in a cloud). Here "better" is in the sense of speedup (local time versus cloud time) and energy.

6.12 The analysis of Sect. 6.10 allows one to get an idea of performance tradeoffs using just mathematical analysis. It does not negate the need for experimental work to confirm performance predictions and find factors that are important that were not considered in the initial analysis. The analysis can also guide the setup of experimental work.

7

Solutions

Problem 7.1:

a. Supervised learning
b. Supervised learning

c. Reinforcement learning
d. Supervised learning
e. Unsupervised learning

Problem 7.2:
Test dataset: 0.3 * 12,000 = 3600 samples
Validation dataset: 0.7 * 12,000 * (1/4) = 2100 samples
Training dataset: 0.7 * 12,000 * (3/4) = 6300 samples.

Problem 7.3:

		Predicted	
		Yes	No
Truth	Yes	4	3
	No	2	1

Problem 7.4:
Accuracy = (4 + 2)/10 = 0.6
Recall = 4/(4 + 3) = 0.57
Precision = 4/(4 + 2) = 0.67
F1 score = 2 * 0.57 * 0.67/(0.57 + 0.67) = 0.61

Problem 7.5:
Log loss is essentially the sum of the probability (output of the model) of the true class of each sample.
Log
loss = -–1/10 * (log(0.8) + log(0.6) + log(0.7) + log(0.5) + log(0.7) + log(0.8) + log(0.5) + log(0.8) + log(0.9) + log(0.9))
= 0.1515
Problem 7.6:
MAE = 17
MSE = 425
MAPE = 0.076
RMSE = 20.61

Problem 7.7:

	G_1	G_2	G_3	
C_1	2	0	1	3
C_2	1	1	0	2

C_3	0	0	1	1
	3	1	2	

Problem 7.8[SD4]:

Rand index A(d_i,d_j) table	1	2	3	4	5	6
1		1	0	0	1	1
2			0	0	1	1
3				1	1	0
4					0	1
5						1
6						

RI = sum of A(di,dj)/(k,2) = 9/15 = 0.6

Problem 7.9:
FMI TP, FP, TN, FN table.

	1	2	3	4	5	6
1		TP	FP	FN	TN	TN
2			FP	FN	TN	TN
3				TN	TN	FP
4					FP	TN
5						TN
6						

TP = 1
FP = 4
TN = 8
FN = 2
FMI = sqrt(1/(1 + 4) * 1/(1 + 3)) = 0.2236

Problem 10:
Hidden layer neuron:

$$\begin{pmatrix} 3 & 5 & 1 & 6 \\ 4 & 1 & 2 & 2 \end{pmatrix} * \begin{pmatrix} 2 & 0 & 3 \\ 1 & 1 & 2 \\ 2 & 0 & 1 \\ 3 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 31 & 11 & 38 \\ 19 & 3 & 22 \end{pmatrix}$$

Output layer:

$$\begin{pmatrix} 31 & 11 & 38 \\ 19 & 3 & 22 \end{pmatrix} * \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 1 \end{pmatrix} = \begin{pmatrix} 254 & 144 \\ 138 & 72 \end{pmatrix}$$

Note ReLU activation function (i.e., f(x) = max(x,0)) has no effect here because all numbers are positive.