Simulação de Evento Discreto

Simulação de evento discreto

- As variáveis de estado modificam-se apenas pela ocorrência de eventos
- Os eventos ocorrem instantaneamente em pontos separados no tempo
- São simulados de forma eficiente
- Uma grande variedade de sistemas do mundo real são bem modelados por evento discreto

Mecanismos de avanço de tempo

- A variável tempo (clock) determina o andamento da simulação (modelo dinâmico)
- Tempo simulado X Tempo real
- A unidade de tempo não é relevante, mas todas as variáveis devem utilizar a mesma
- Propostas para avanço de tempo
  - Avanço de tempo pelo próximo evento
    - Avança para o instante do próximo evento
    - Mais utilizado (por que?)
  - Avanço de tempo por incremento fixo
    - Tempo discretizado. Em cada ponto verifica-se eventos.

Mecanismos de avanço de tempo

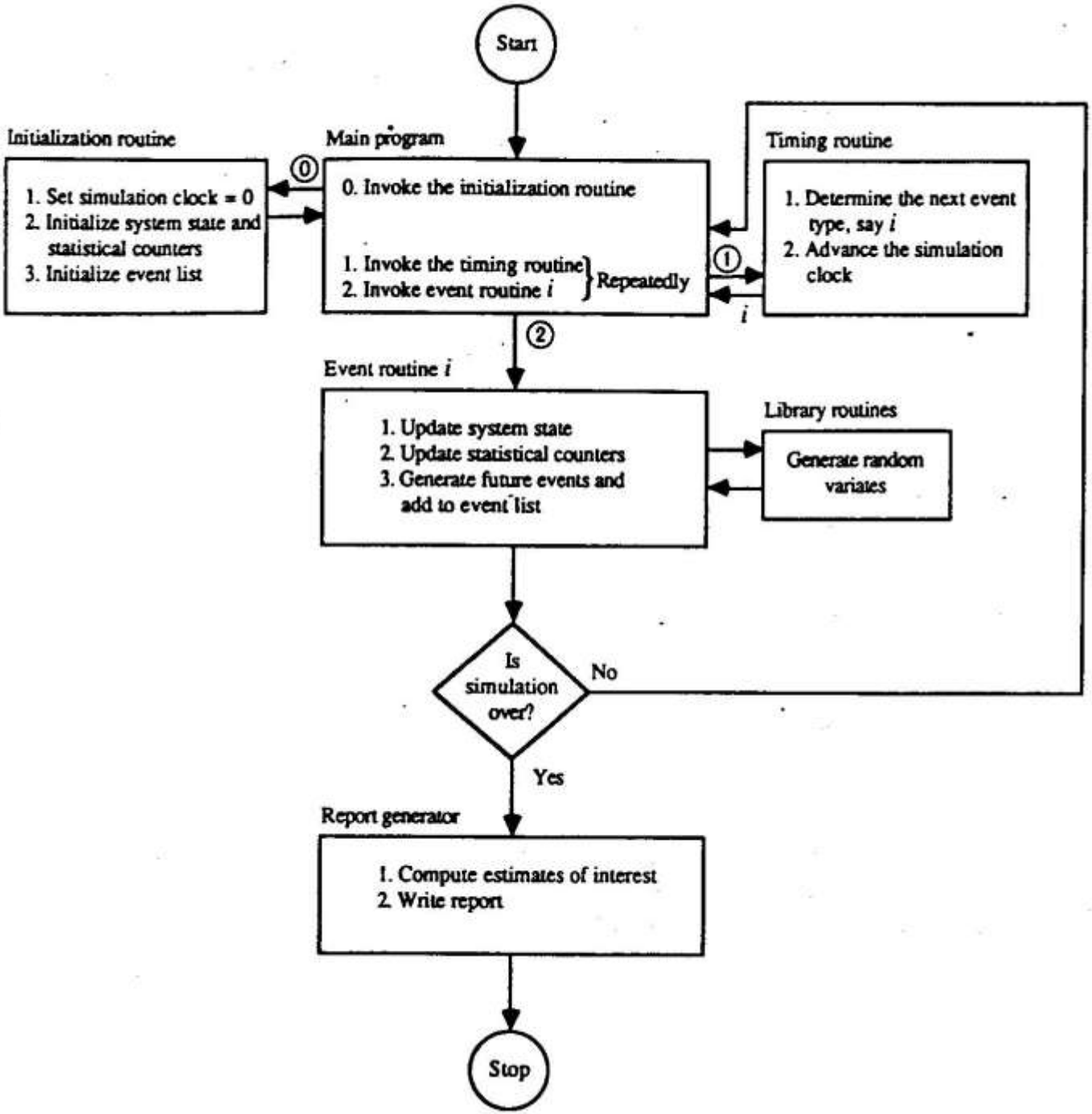
- Como não ocorre modificações no sistema entre eventos, o avanço para o próximo evento salta períodos de inatividade
  - Por outro lado, o avanço por incremento fixo consome muita CPU desnecessariamente

Componentes de uma simulação de evento discreto

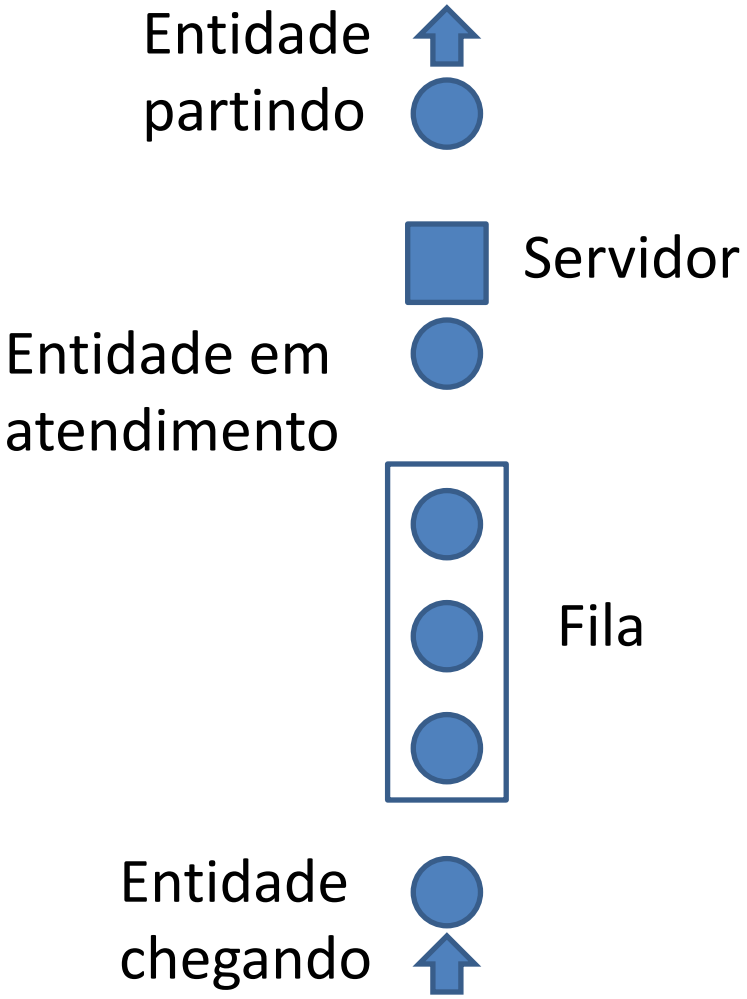
- Variáveis de estado
- Clock
- Lista de eventos: contém o instante do próximo evento de cada tipo
- Contadores estatísticos: performance do sistema
- Rotina de inicialização (instante zero)
- Rotina de tempo: determina o próximo evento e avança o clock
- Rotina de evento (uma por tipo): atualiza o sistema quando um dado evento ocorre

Componentes de uma simulação de evento discreto

- Rotinas de biblioteca: geradores de variáveis aleatórias
- Gerador de relatório (final): calcula estimativas das medidas de performance (com base nos contadores) e imprime
- Principal: repetidamente chama a rotina de tempo e transfere controle para a rotina de evento correspondente.
  - Também chama inicialização e geração de relatório


Start
Initialization routine
1. Set simulation clock = 0
2. Initialize system state and statistical counters
3. Initialize event list
Main program
0. Invoke the initialization routine
1. Invoke the timing routine
2. Invoke event routine i
Repeatedly
Timing routine
1. Determine the next event type, say i
2. Advance the simulation clock
Event routine i
1. Update system state
2. Update statistical counters
3. Generate future events and add to event list
Library routines
Generate random variates
Is simulation over?
No
Yes
Report generator
1. Compute estimates of interest
2. Write report
Stop

Simulação de sistema de fila com um servidor



- Tempo entre chegadas e tempo de atendimento: duas v.a. exponenciais i.i.d.
- Sistema inicialmente vazio: fila e servidor
- Fim da simulação: n-ésima entidade entra em atendimento
  - Ou um evento de fim de simulação

Simulação de sistema de fila com um servidor

- Medidas de performance:
  - Estimador do atraso médio na fila *d(n):*

$$\hat{d}(n) = \frac{\sum_{i=1}^{n} D_i}{n}$$

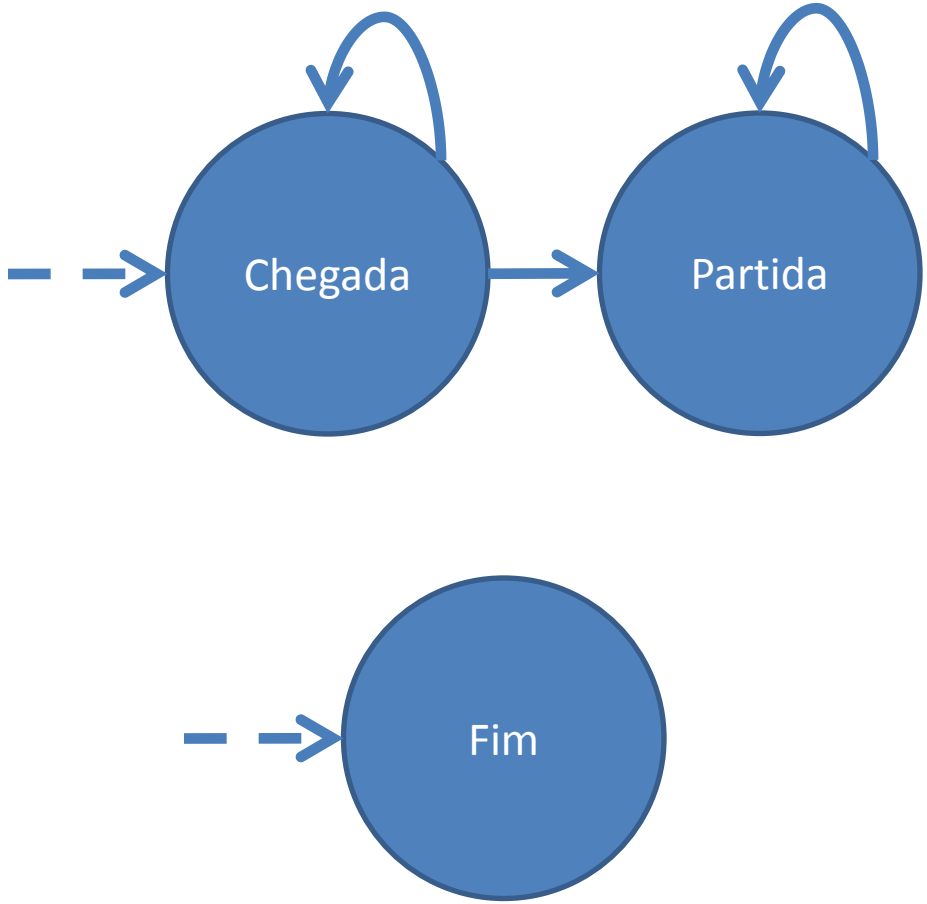
  - Estimador da *média no tempo* do número de clientes na fila *q(n)*:
    - Pi: proporção do tempo com i clientes

$$\hat{q}(n) = \sum_{i=0}^{\infty} i\hat{p}_i = \frac{\sum_{i=0}^{n} iT_i}{T(n)} = \frac{\int_0^{T(n)} Q(t)dt}{T(n)}$$

  - Utilização esperada: proporção esperada do tempo em que o servidor está ocupado

$$\hat{u}(n) = \frac{\int_0^{T(n)} B(t)dt}{T(n)}$$

Simulação de sistema de fila com um servidor: Eventos



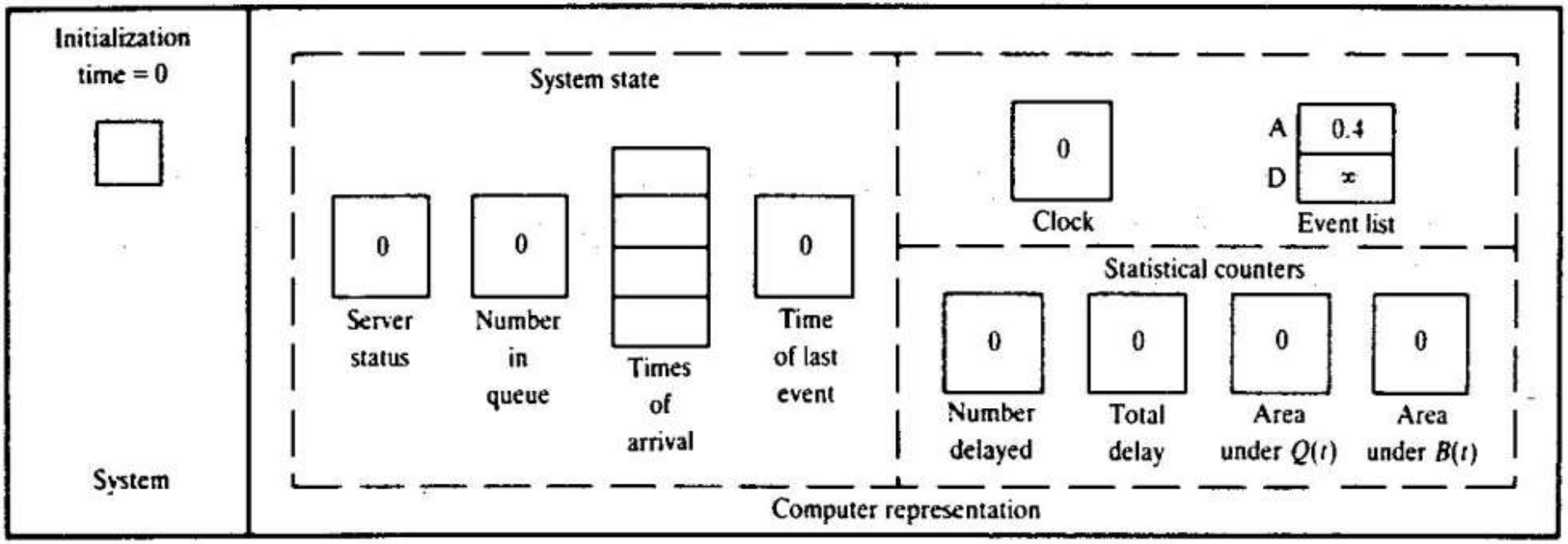
- Cada componente conexa deve ter pelo menos um evento na inicialização

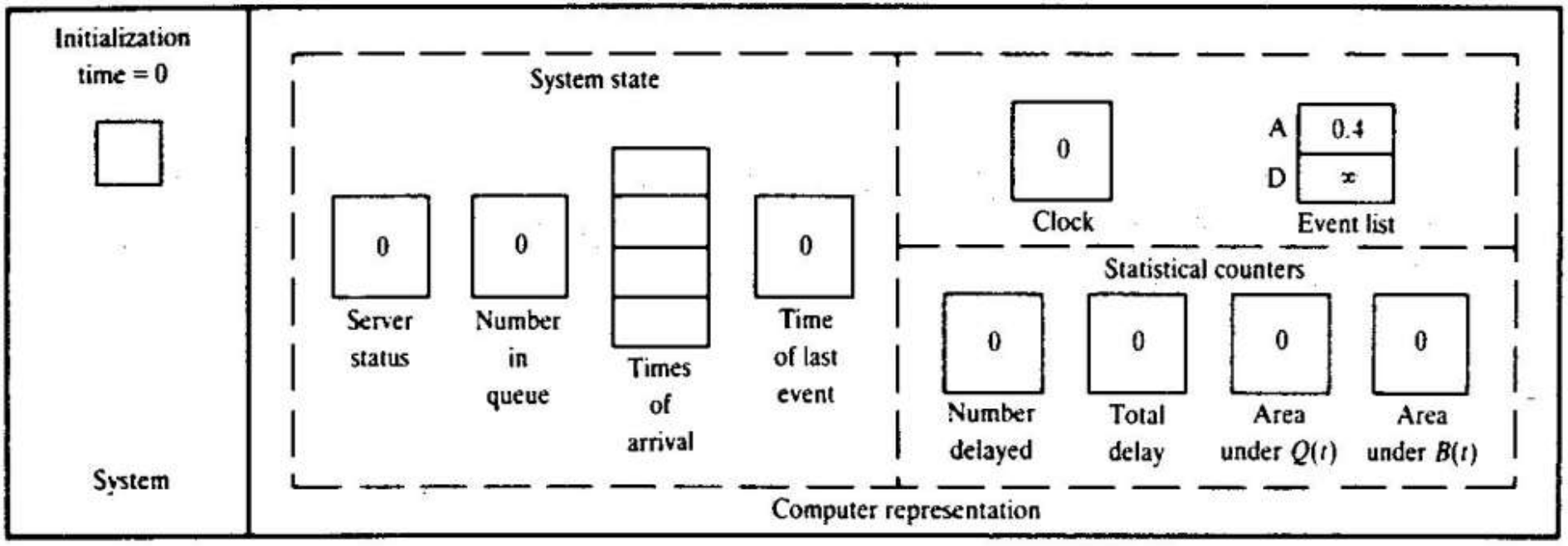
Rotina do evento CHEGADA

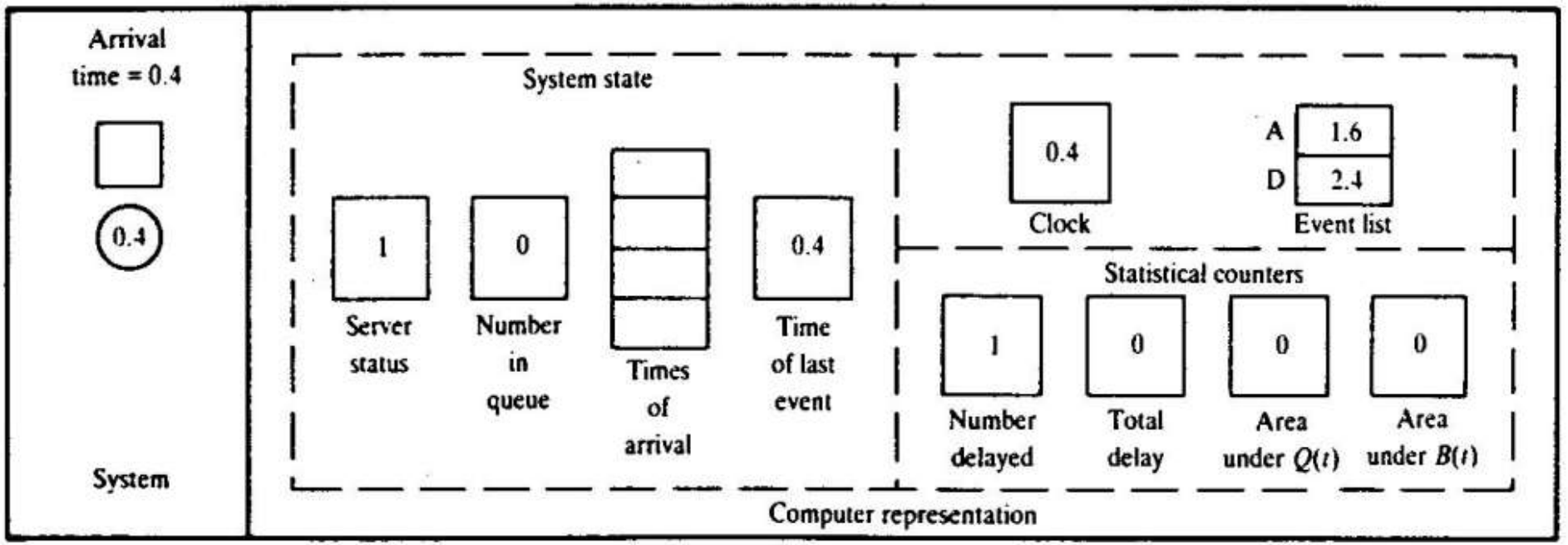
- Escalone o próximo evento de chegada
- Se o servidor está ocupado
  - Incremente contador de elementos na fila
  - Se fila cheia, escreva msg de erro e encerre
  - Armaze o instante de chegada do cliente
- Senão
  - Registre atraso zero para este cliente
  - Incremente o número de clientes atendidos
  - Marque o servidor como ocupado
  - Escalone a partida deste cliente

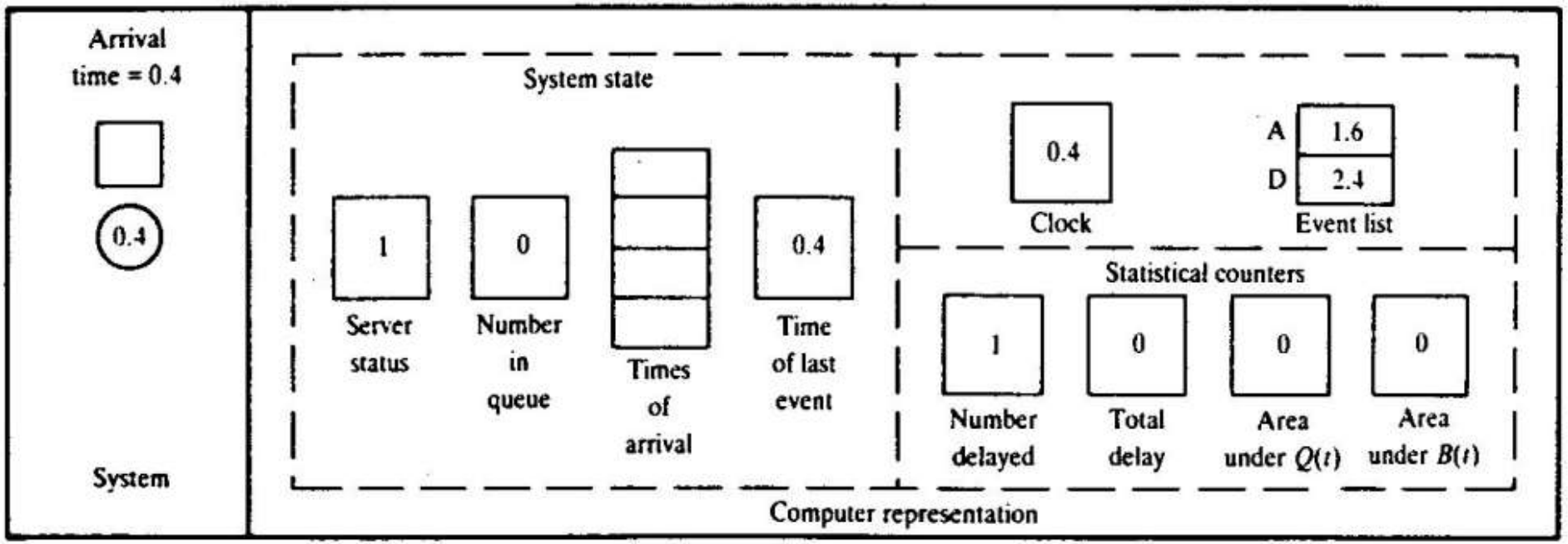
Rotina do evento PARTIDA

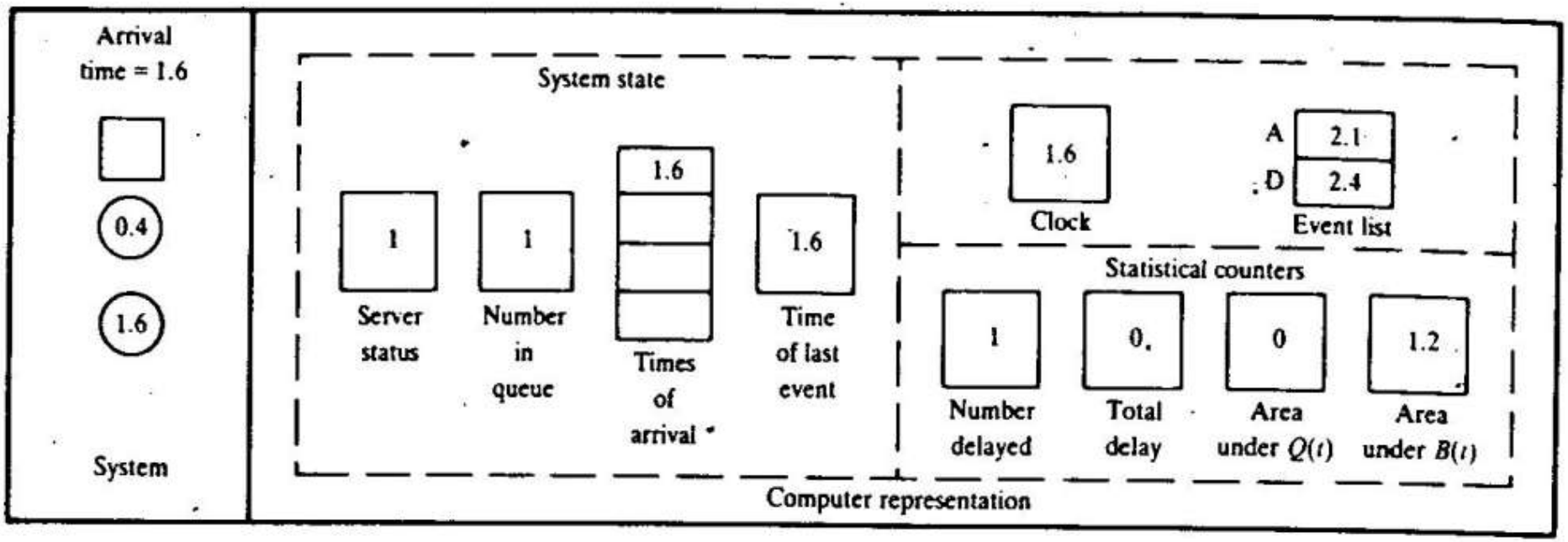
- Se a fila está vazia
  - Marque o servidor como ocioso
  - Indique que não existe próximo evento de partida (tempo infinito)
- Senão
  - Decremente contador de elementos na fila
  - Remova o cliente da fila
  - Calcule o atrado do cliente que entrou em serviço
  - Incremente o número de clientes atendidos
  - Escalone o evento de partida deste cliente

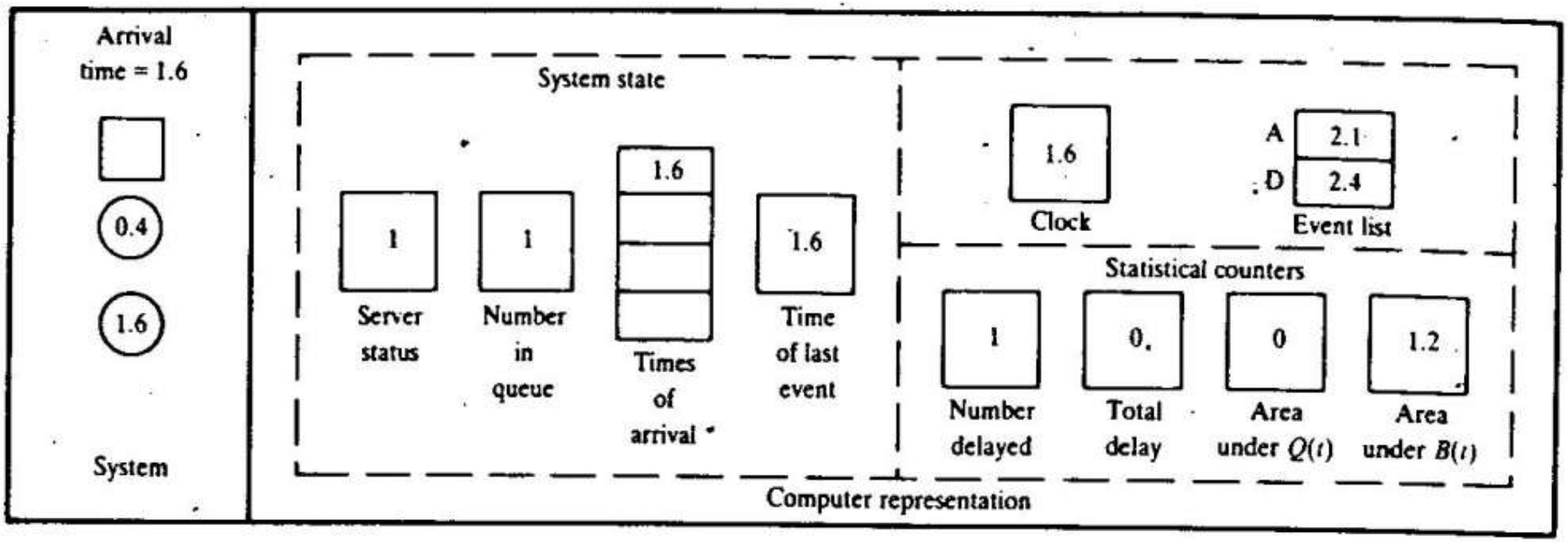

Initialization
time = 0
System
System state
0
Server status
0
Number in queue
Times of arrival
0
Time of last event
0
Clock
A
0.4
D
∞
Event list
Statistical counters
0
Number delayed
0
Total delay
0
Area under $Q(t)$
0
Area under $B(t)$
Computer representation

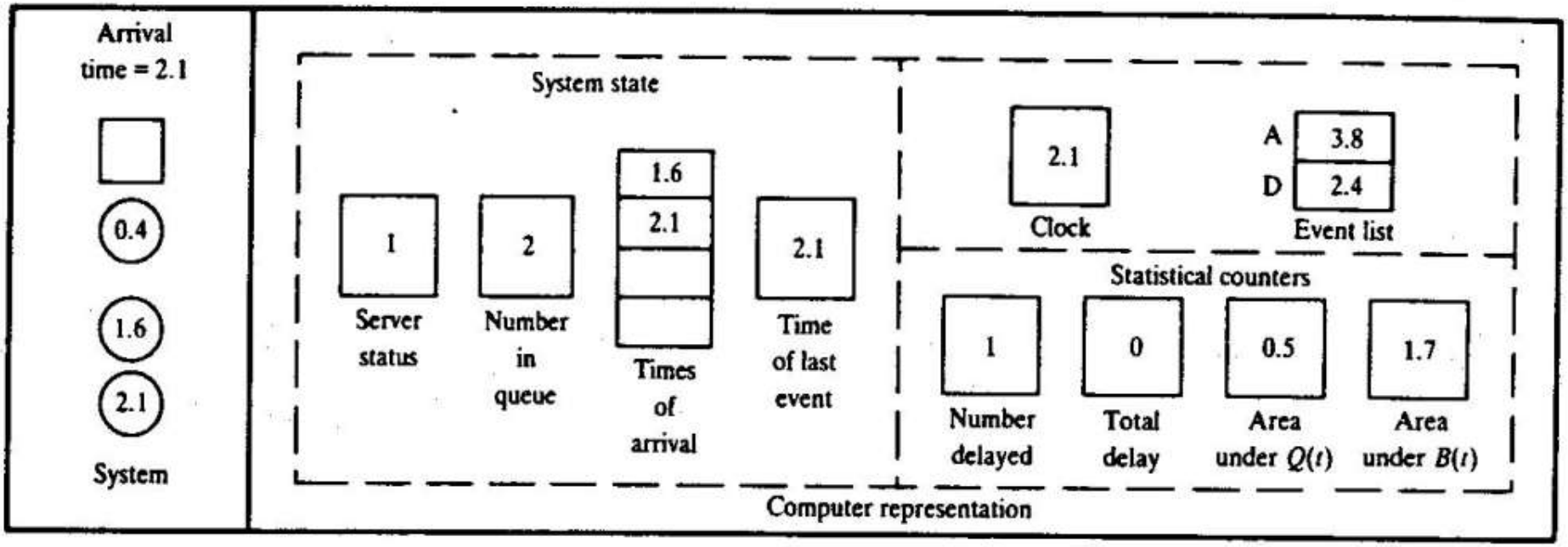

Initialization
time = 0
System
System state
0
Server status
0
Number in queue
Times of arrival
0
Time of last event
0
Clock
A
D
0.4
∞
Event list
Statistical counters
0
Number delayed
0
Total delay
0
Area under Q(t)
0
Area under B(t)
Computer representation

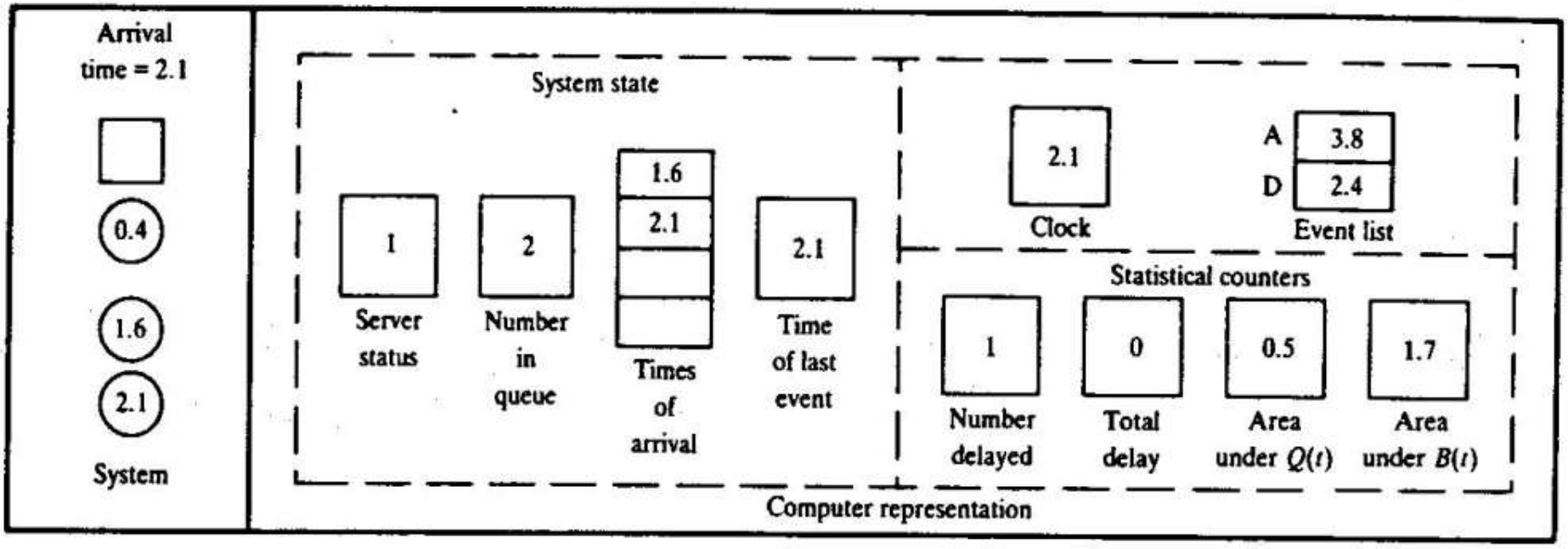

Arrival
time = 0.4
0.4
System
System state
1
Server status
0
Number in queue
Times of arrival
0.4
Time of last event
0.4
Clock
A
D
1.6
2.4
Event list
Statistical counters
1
Number delayed
0
Total delay
0
Area under Q(t)
0
Area under B(t)
Computer representation

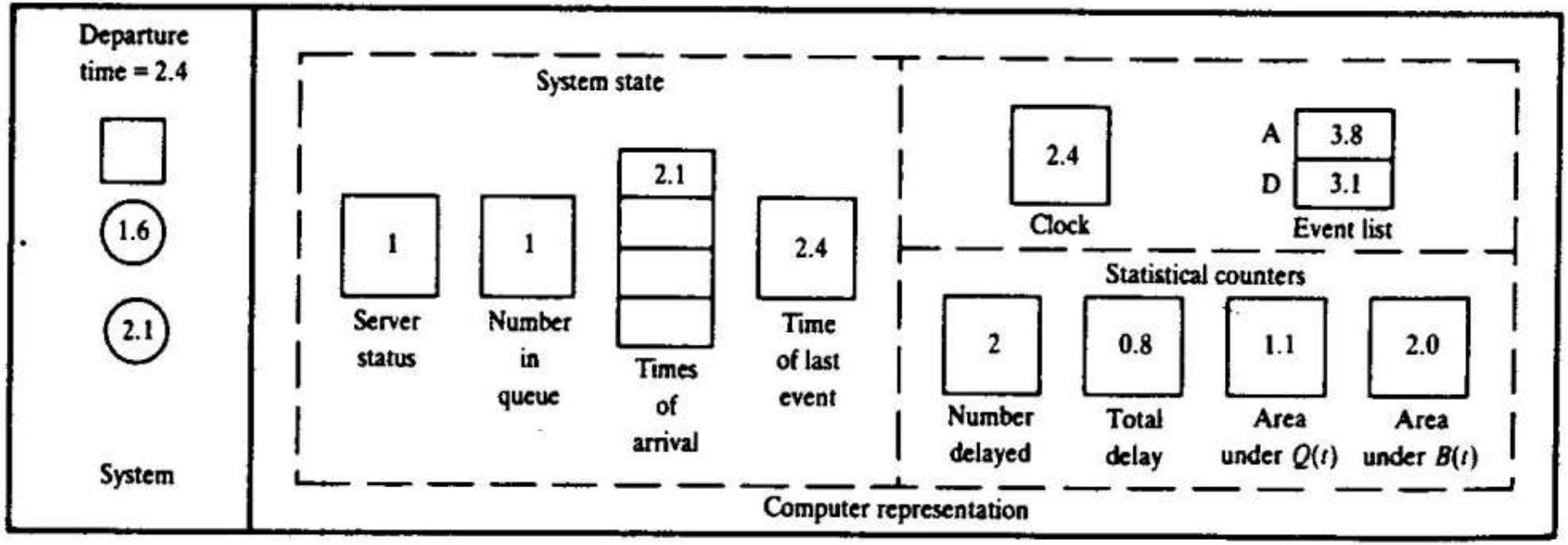

Arrival
time = 0.4
0.4
System
System state
1
Server status
0
Number in queue
Times of arrival
0.4
Time of last event
0.4
Clock
A
1.6
D
2.4
Event list
Statistical counters
1
Number delayed
0
Total delay
0
Area under $Q(t)$
0
Area under $B(t)$
Computer representation

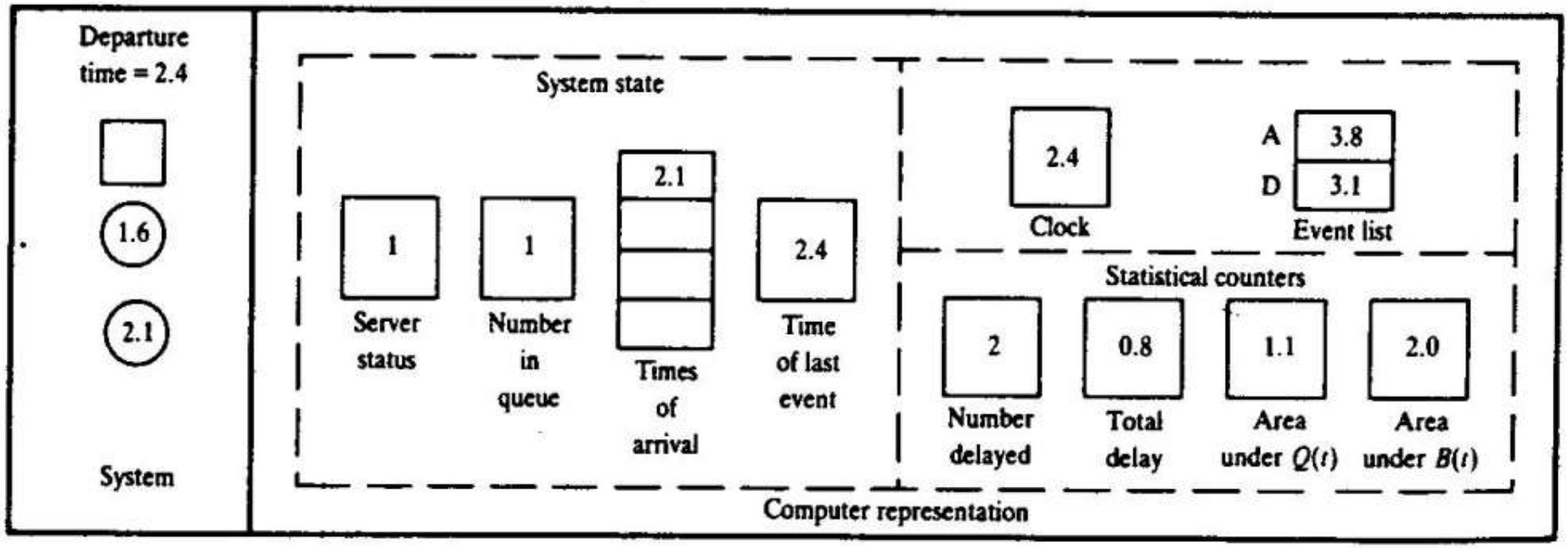

Arrival
time = 1.6
0.4
1.6
System
System state
1
Server status
1
Number in queue
1.6
Times of arrival
1.6
Time of last event
1.6
Clock
A
2.1
D
2.4
Event list
Statistical counters
1
Number delayed
0.
Total delay
0
Area under $Q(t)$
1.2
Area under $B(t)$
Computer representation

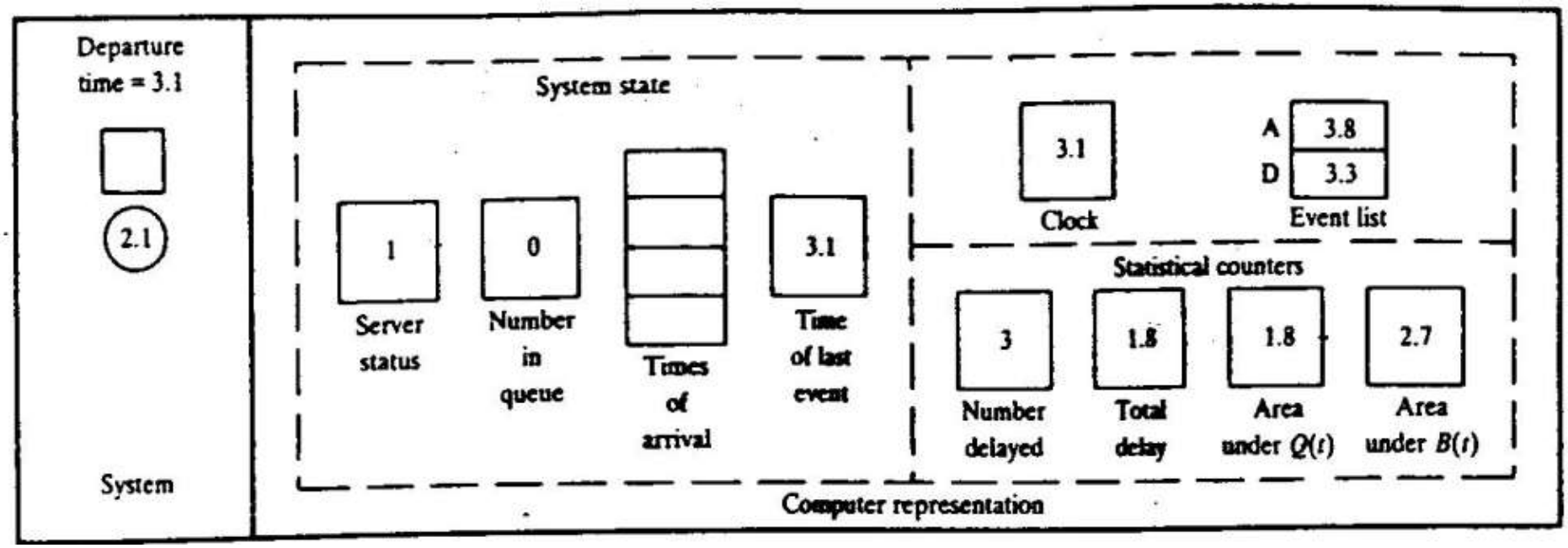

Arrival
time = 1.6
0.4
1.6
System
System state
1
Server status
1
Number in queue
1.6
Times of arrival
1.6
Time of last event
1.6
Clock
A 2.1
D 2.4
Event list
Statistical counters
1
Number delayed
0.
Total delay
0
Area under Q(t)
1.2
Area under B(t)
Computer representation

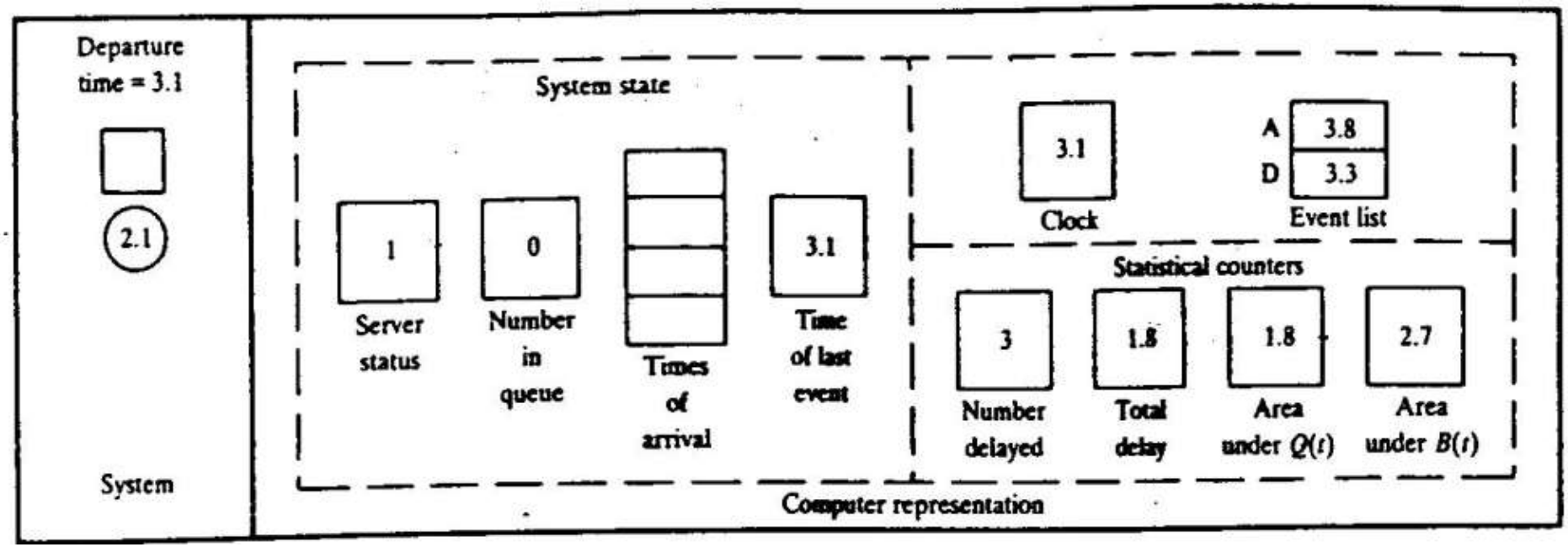

Arrival
time = 2.1
0.4
1.6
2.1
System
System state
1
Server status
2
Number in queue
1.6
2.1
Times of arrival
2.1
Time of last event
2.1
Clock
A 3.8
D 2.4
Event list
Statistical counters
1
Number delayed
0
Total delay
0.5
Area under Q(t)
1.7
Area under B(t)
Computer representation

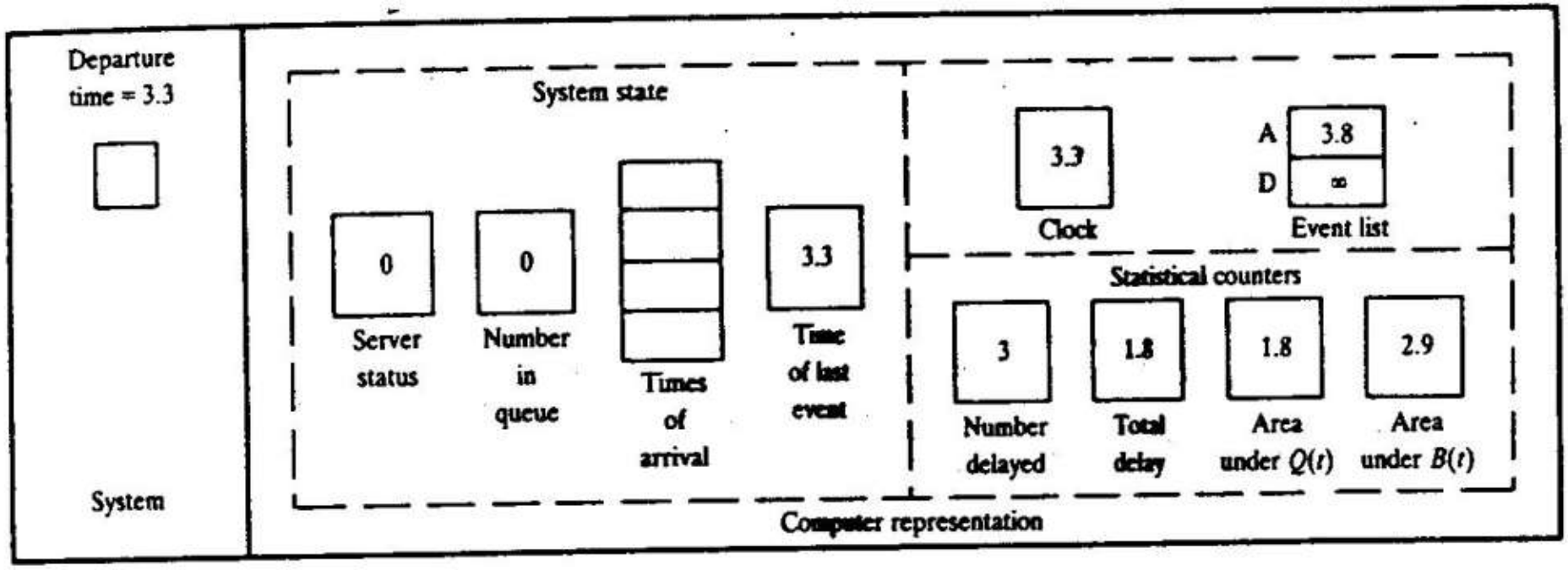

Arrival
time = 2.1
0.4
1.6
2.1
System
System state
1
Server status
2
Number in queue
1.6
2.1
Times of arrival
2.1
Time of last event
2.1
Clock
A 3.8
D 2.4
Event list
Statistical counters
1
Number delayed
0
Total delay
0.5
Area under $Q(t)$
1.7
Area under $B(t)$
Computer representation

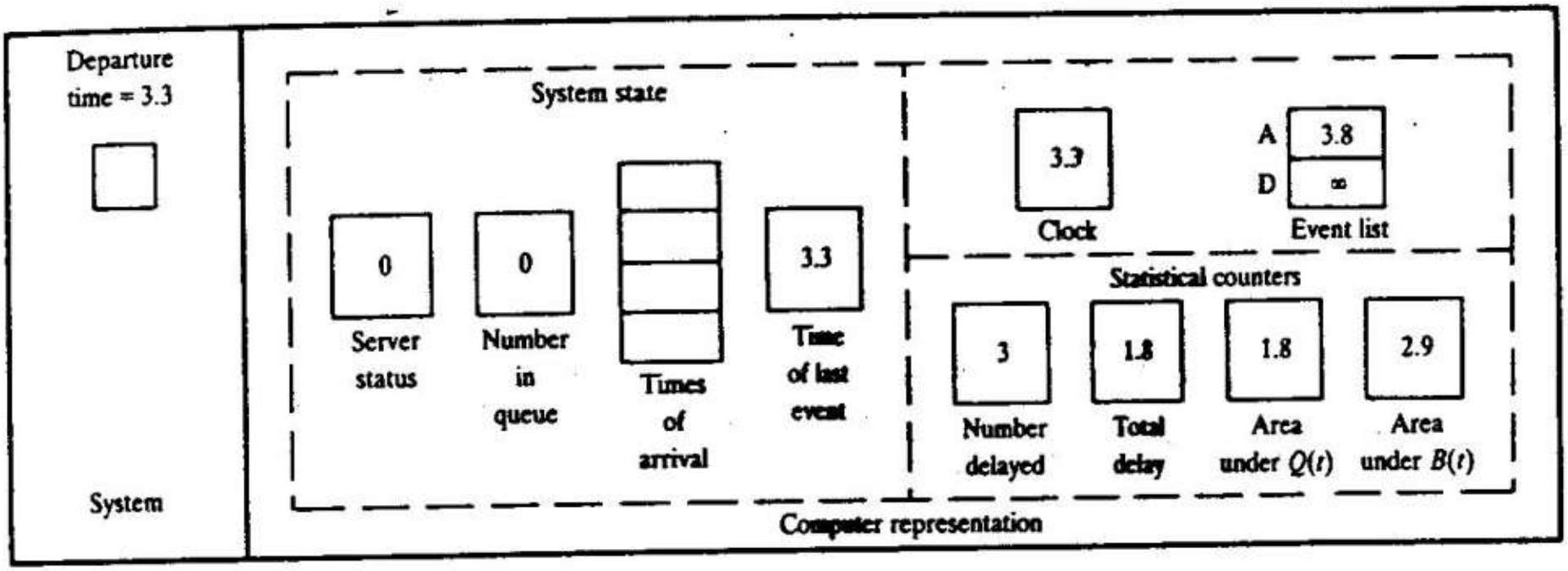

Departure
time = 2.4
1.6
2.1
System
System state
1
Server status
1
Number in queue
2.1
Times of arrival
2.4
Time of last event
2.4
Clock
A 3.8
D 3.1
Event list
Statistical counters
2
Number delayed
0.8
Total delay
1.1
Area under $Q(t)$
2.0
Area under $B(t)$
Computer representation

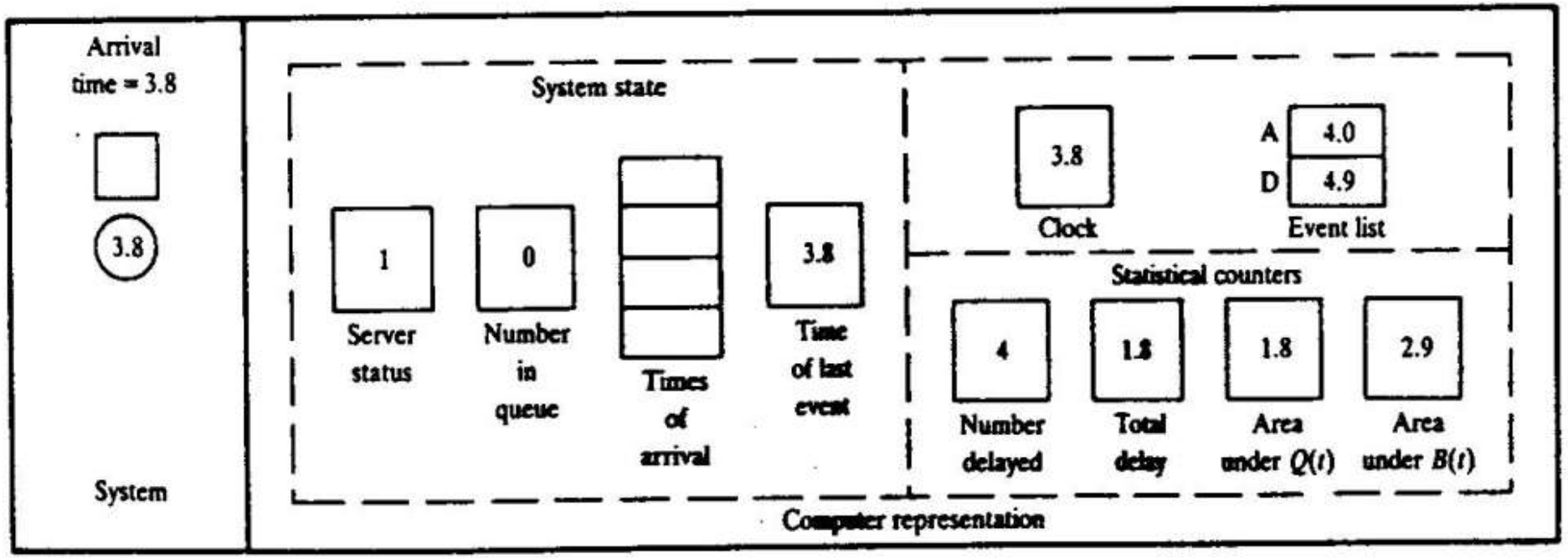

Departure
time = 2.4
1.6
2.1
System
System state
1
Server
status
1
Number
in
queue
2.1
Times
of
arrival
2.4
Time
of last
event
2.4
Clock
A
D
3.8
3.1
Event list
Statistical counters
2
Number
delayed
0.8
Total
delay
1.1
Area
under $Q(t)$
2.0
Area
under $B(t)$
Computer representation

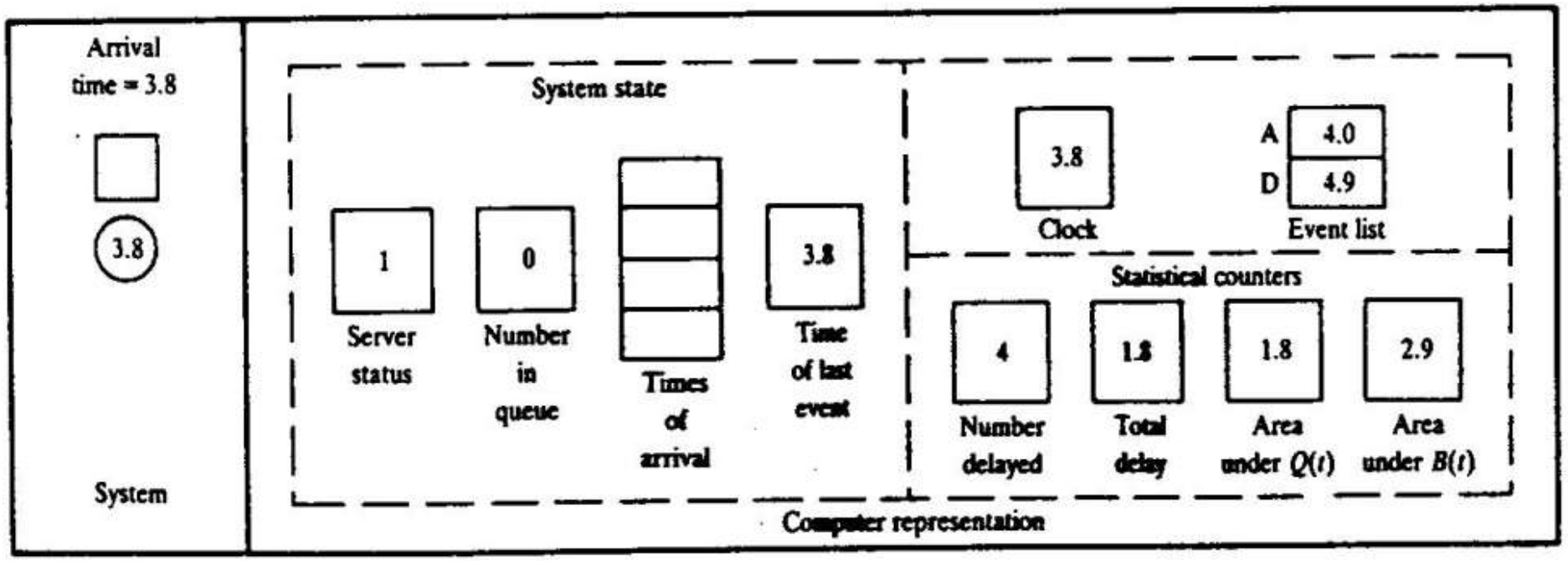

Departure
time = 3.1
2.1
System
System state
1
Server
status
0
Number
in
queue
Times
of
arrival
3.1
Time
of last
event
3.1
Clock
A
D
3.8
3.3
Event list
Statistical counters
3
Number
delayed
1.8
Total
delay
1.8
Area
under $Q(t)$
2.7
Area
under $B(t)$
Computer representation

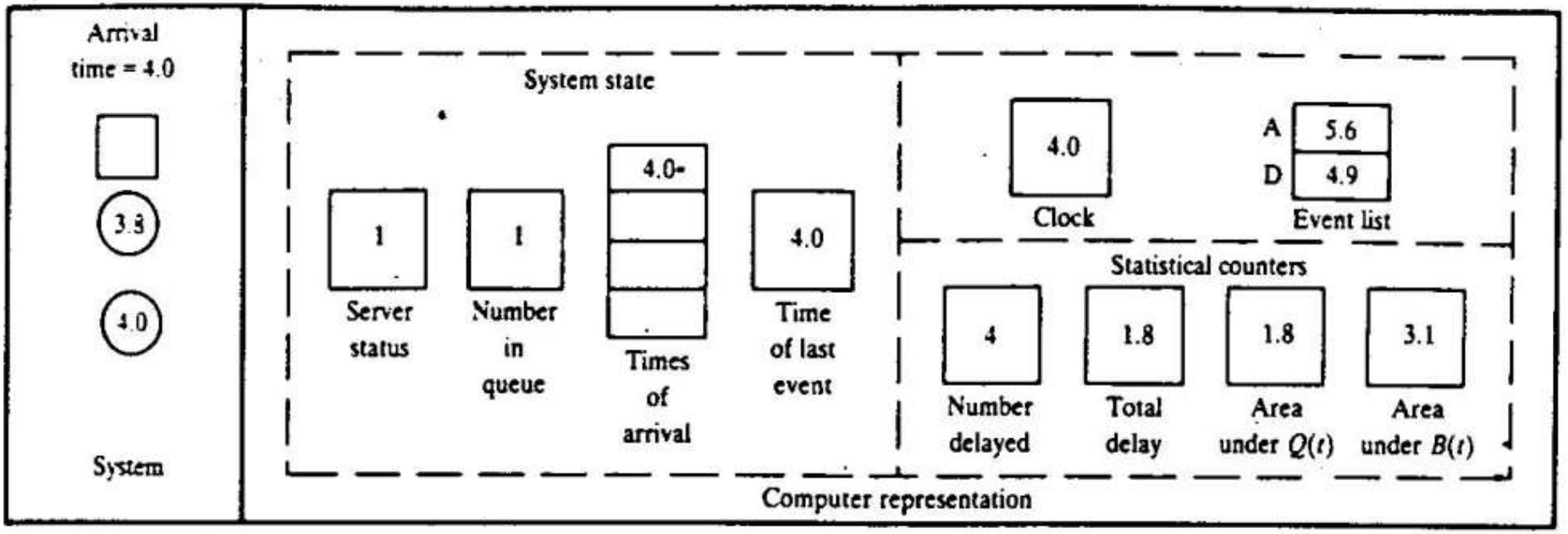

Departure
time = 3.1
2.1
System
System state
1
Server
status
0
Number
in
queue
Times
of
arrival
3.1
Time
of last
event
3.1
Clock
A
D
3.8
3.3
Event list
Statistical counters
3
Number
delayed
1.8
Total
delay
1.8
Area
under $Q(t)$
2.7
Area
under $B(t)$
Computer representation

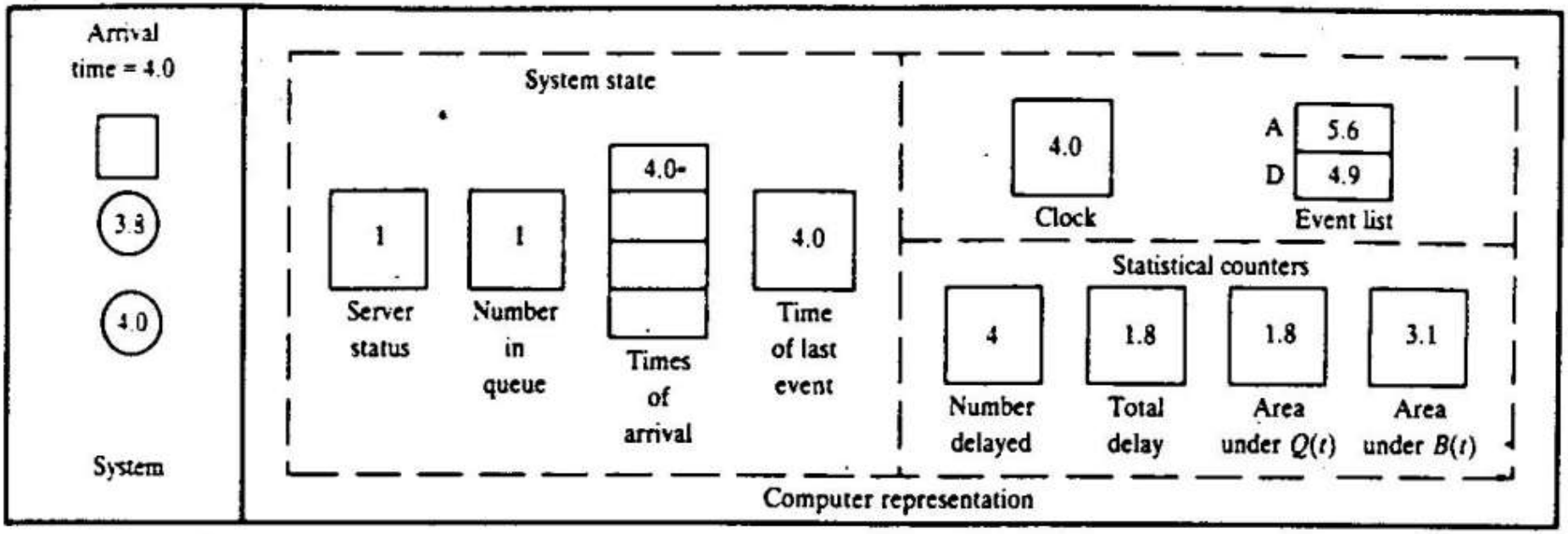

Departure
time = 3.3
System
System state
0
Server
status
0
Number
in
queue
Times
of
arrival
3.3
Time
of last
event
3.3
Clock
A
D
3.8
∞
Event list
Statistical counters
3
Number
delayed
1.8
Total
delay
1.8
Area
under $Q(t)$
2.9
Area
under $B(t)$
Computer representation

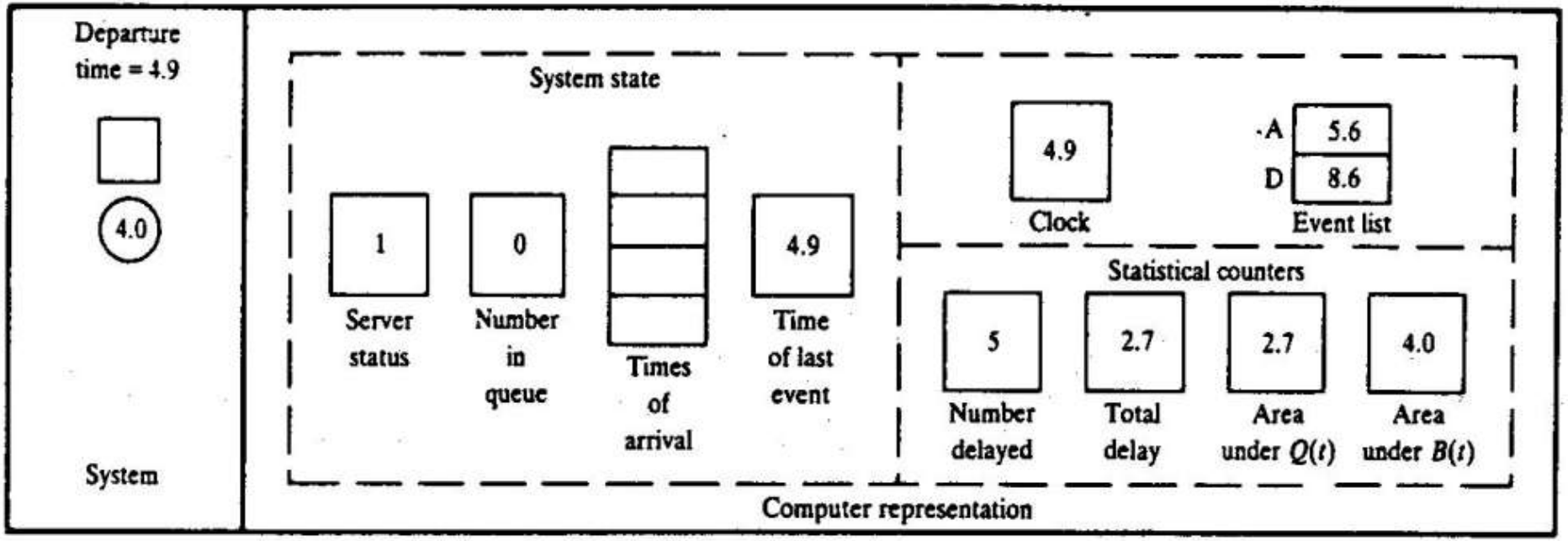

Departure
time = 3.3
System
System state
0
Server status
0
Number in queue
Times of arrival
3.3
Time of last event
3.3
Clock
A
D
3.8
∞
Event list
Statistical counters
3
Number delayed
1.8
Total delay
1.8
Area under $Q(t)$
2.9
Area under $B(t)$
Computer representation

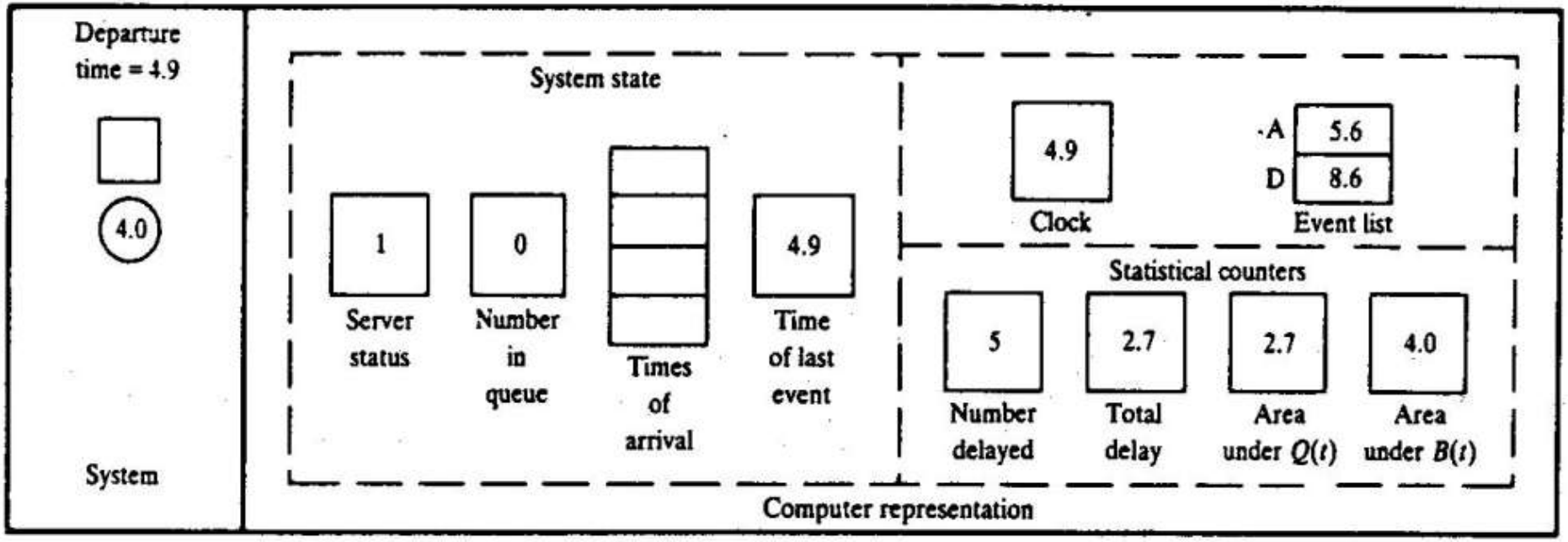

Arrival
time = 3.8
3.8
System
System state
1
Server status
0
Number in queue
Times of arrival
3.8
Time of last event
3.8
Clock
A
D
4.0
4.9
Event list
Statistical counters
4
Number delayed
1.8
Total delay
1.8
Area under $Q(t)$
2.9
Area under $B(t)$
Computer representation

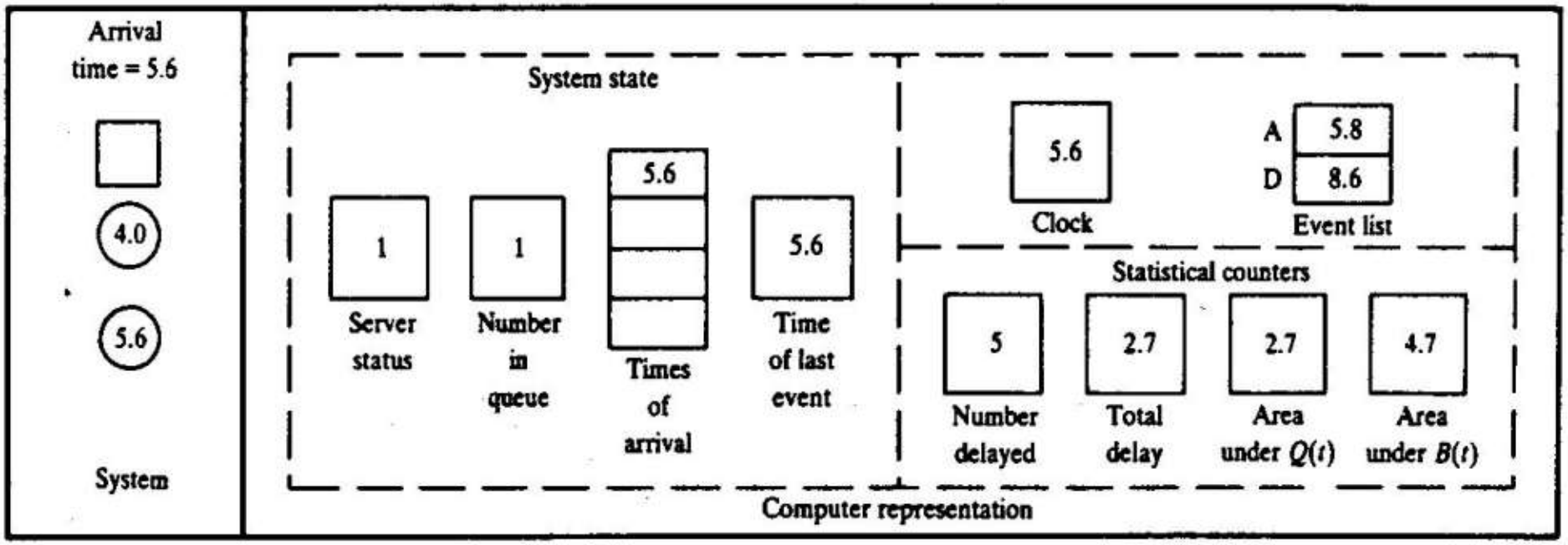

Arrival
time = 3.8
3.8
System
System state
1
Server status
0
Number in queue
Times of arrival
3.8
Time of last event
3.8
Clock
A
4.0
D
4.9
Event list
Statistical counters
4
Number delayed
1.8
Total delay
1.8
Area under Q(t)
2.9
Area under B(t)
Computer representation

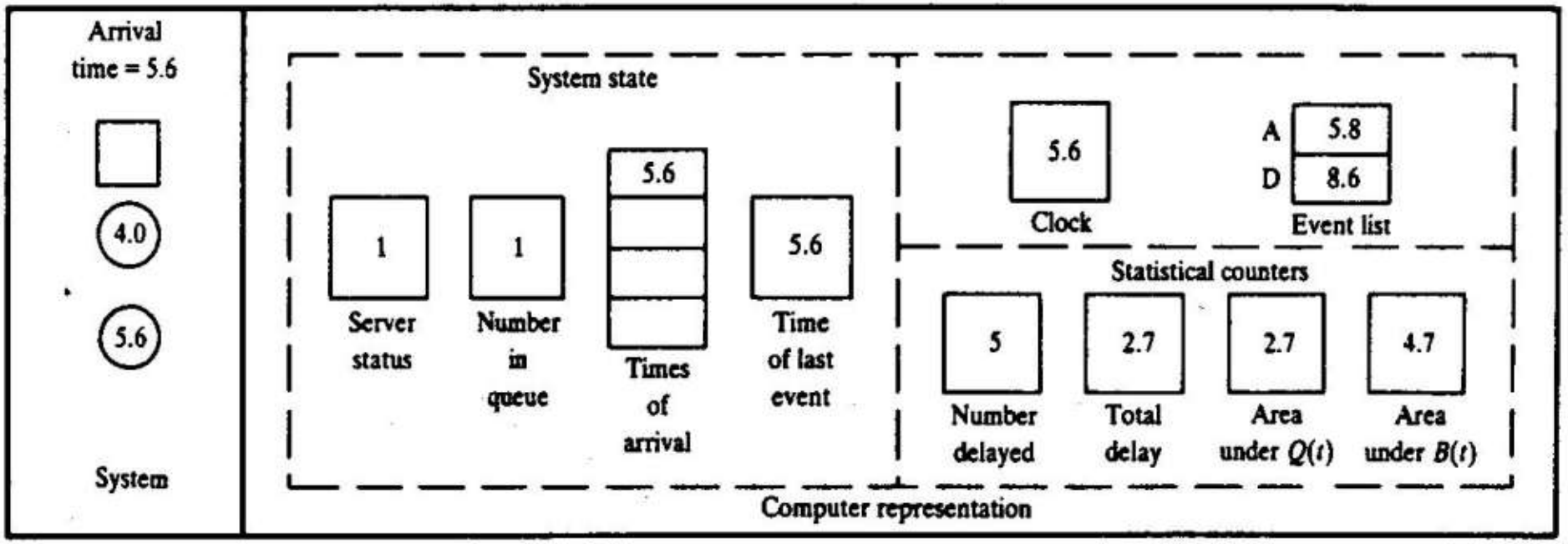

Arrival
time = 4.0
3.8
4.0
System
System state
1
Server status
1
Number in queue
4.0-
Times of arrival
4.0
Time of last event
4.0
Clock
A
5.6
D
4.9
Event list
Statistical counters
4
Number delayed
1.8
Total delay
1.8
Area under Q(t)
3.1
Area under B(t)
Computer representation

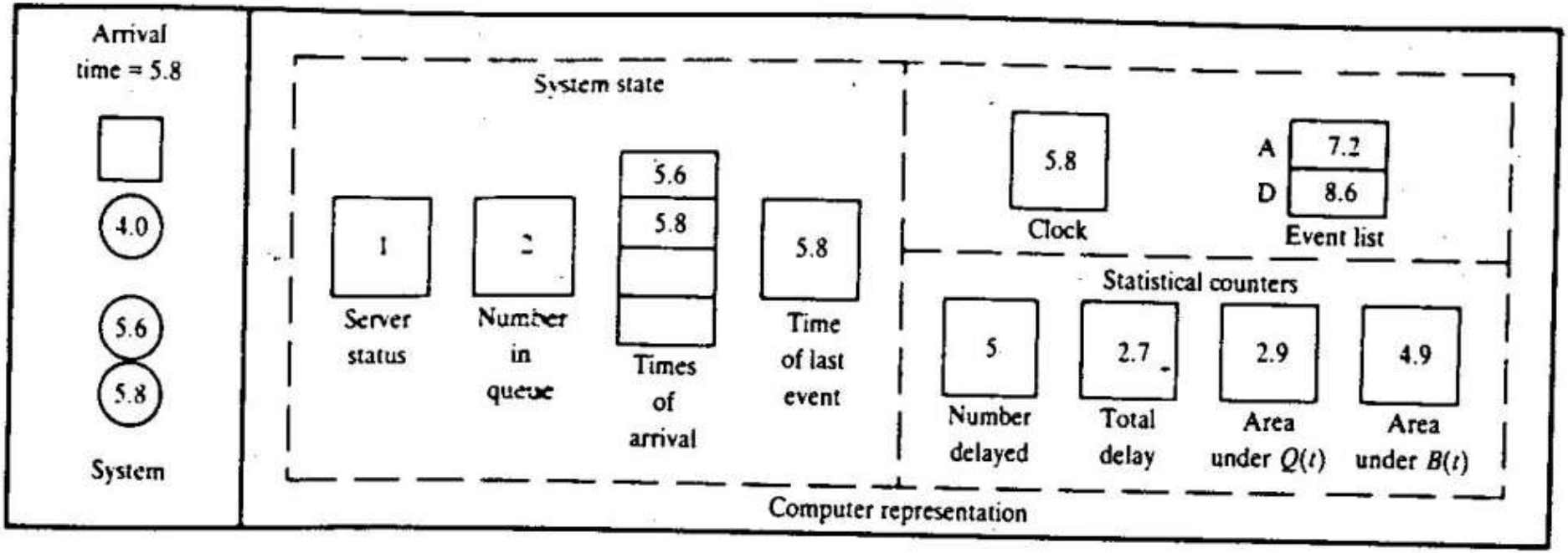

Arrival
time = 4.0
3.3
4.0
System
System state
1
Server
status
1
Number
in
queue
4.0-
Times
of
arrival
4.0
Time
of last
event
4.0
Clock
A
5.6
D
4.9
Event list
Statistical counters
4
Number
delayed
1.8
Total
delay
1.8
Area
under Q(t)
3.1
Area
under B(t)
Computer representation

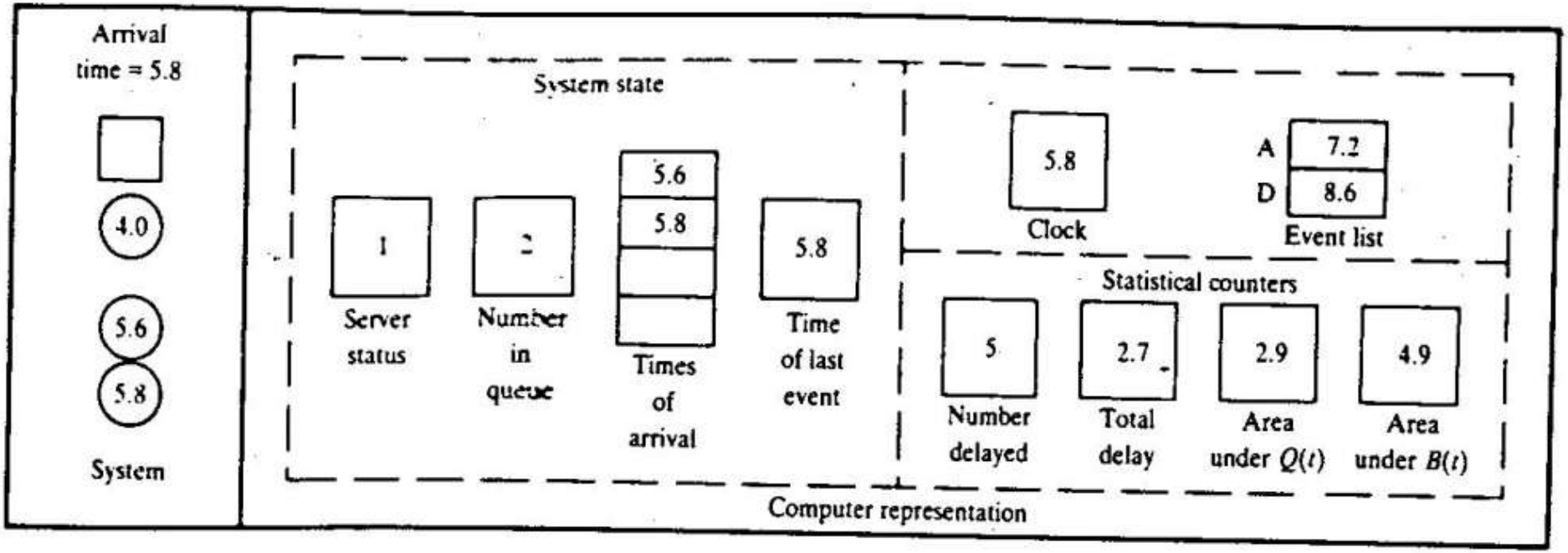

Departure
time = 4.9
4.0
System
System state
1
Server
status
0
Number
in
queue
Times
of
arrival
4.9
Time
of last
event
4.9
Clock
A
5.6
D
8.6
Event list
Statistical counters
5
Number
delayed
2.7
Total
delay
2.7
Area
under Q(t)
4.0
Area
under B(t)
Computer representation

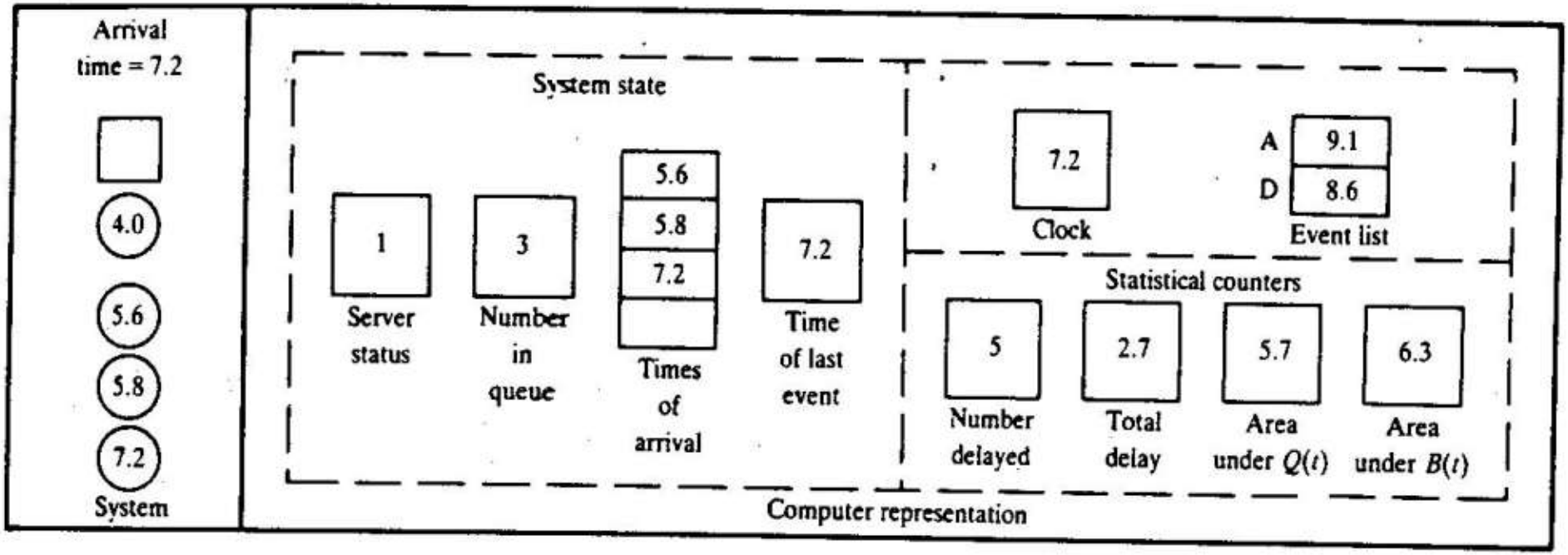

Departure
time = 4.9
4.0
System
System state
1
Server status
0
Number in queue
Times of arrival
4.9
Time of last event
4.9
Clock
A
D
5.6
8.6
Event list
Statistical counters
5
Number delayed
2.7
Total delay
2.7
Area under Q(t)
4.0
Area under B(t)
Computer representation

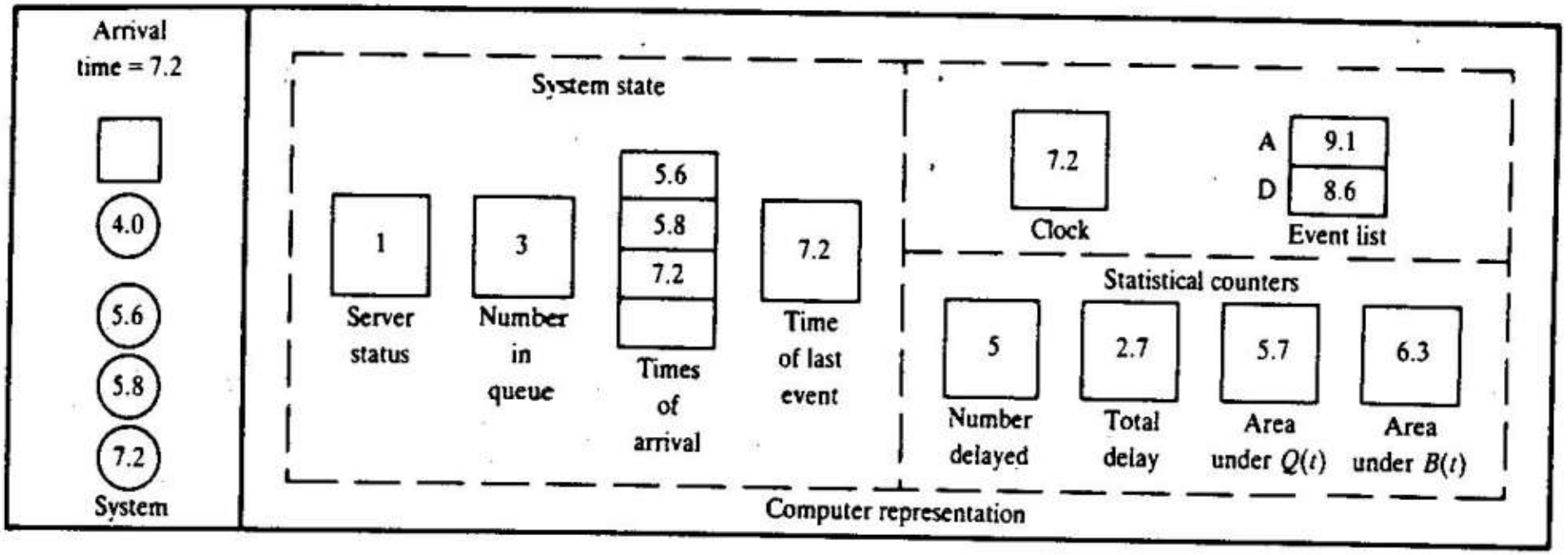

Arrival
time = 5.6
4.0
5.6
System
System state
1
Server status
1
Number in queue
5.6
Times of arrival
5.6
Time of last event
5.6
Clock
A
D
5.8
8.6
Event list
Statistical counters
5
Number delayed
2.7
Total delay
2.7
Area under Q(t)
4.7
Area under B(t)
Computer representation

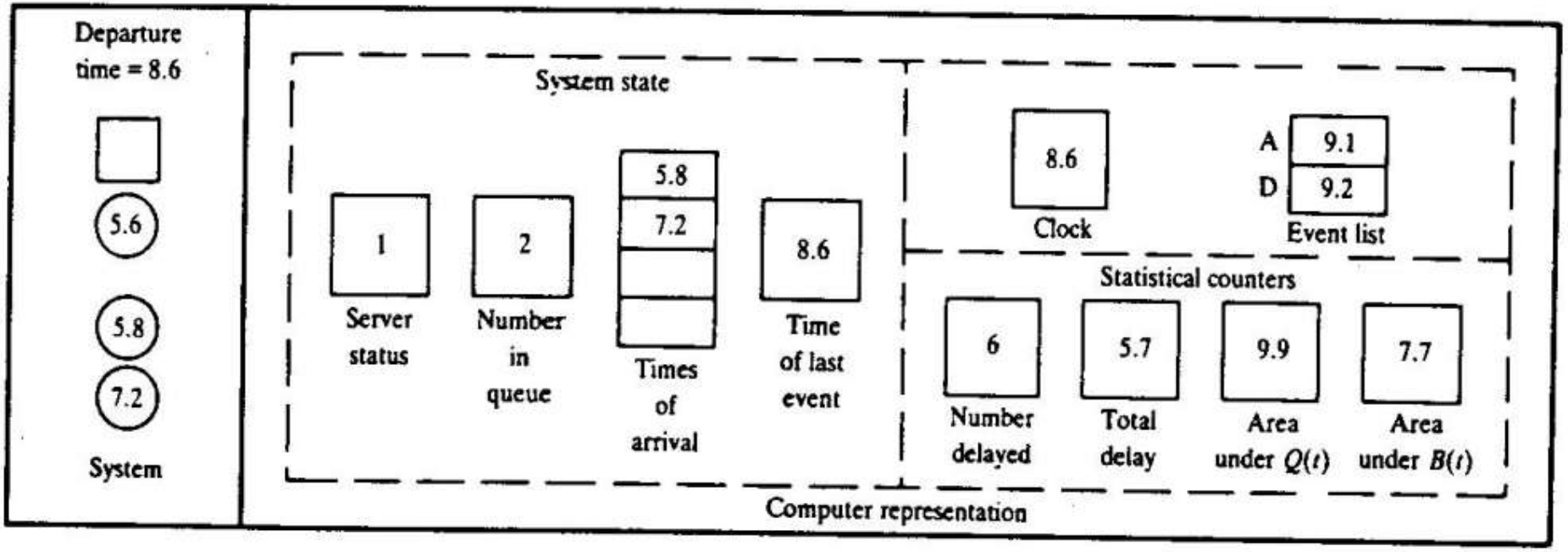

Arrival
time = 5.6
4.0
5.6
System
System state
1
Server status
1
Number in queue
5.6
Times of arrival
5.6
Time of last event
5.6
Clock
A 5.8
D 8.6
Event list
Statistical counters
5
Number delayed
2.7
Total delay
2.7
Area under $Q(t)$
4.7
Area under $B(t)$
Computer representation


Arrival
time = 5.8
4.0
5.6
5.8
System
System state
1
Server status
2
Number in queue
5.6
5.8
Times of arrival
5.8
Time of last event
5.8
Clock
A 7.2
D 8.6
Event list
Statistical counters
5
Number delayed
2.7
Total delay
2.9
Area under $Q(t)$
4.9
Area under $B(t)$
Computer representation


Arrival
time = 5.8
4.0
5.6
5.8
System
System state
1
Server status
2
Number in queue
5.6
5.8
Times of arrival
5.8
Time of last event
5.8
Clock
A
D
7.2
8.6
Event list
Statistical counters
5
Number delayed
2.7
Total delay
2.9
Area under $Q(t)$
4.9
Area under $B(t)$
Computer representation


Arrival
time = 7.2
4.0
5.6
5.8
7.2
System
System state
1
Server status
3
Number in queue
5.6
5.8
7.2
Times of arrival
7.2
Time of last event
7.2
Clock
A
D
9.1
8.6
Event list
Statistical counters
5
Number delayed
2.7
Total delay
5.7
Area under $Q(t)$
6.3
Area under $B(t)$
Computer representation


Arrival
time = 7.2
4.0
5.6
5.8
7.2
System
System state
1
Server status
3
Number in queue
5.6
5.8
7.2
Times of arrival
7.2
Time of last event
7.2
Clock
A 9.1
D 8.6
Event list
Statistical counters
5
Number delayed
2.7
Total delay
5.7
Area under $Q(t)$
6.3
Area under $B(t)$
Computer representation


Departure
time = 8.6
5.6
5.8
7.2
System
System state
1
Server status
2
Number in queue
5.8
7.2
Times of arrival
8.6
Time of last event
8.6
Clock
A 9.1
D 9.2
Event list
Statistical counters
6
Number delayed
5.7
Total delay
9.9
Area under $Q(t)$
7.7
Area under $B(t)$
Computer representation

Simulação de sistema de fila com um servidor: implementação

- Código no site do livro [Law]

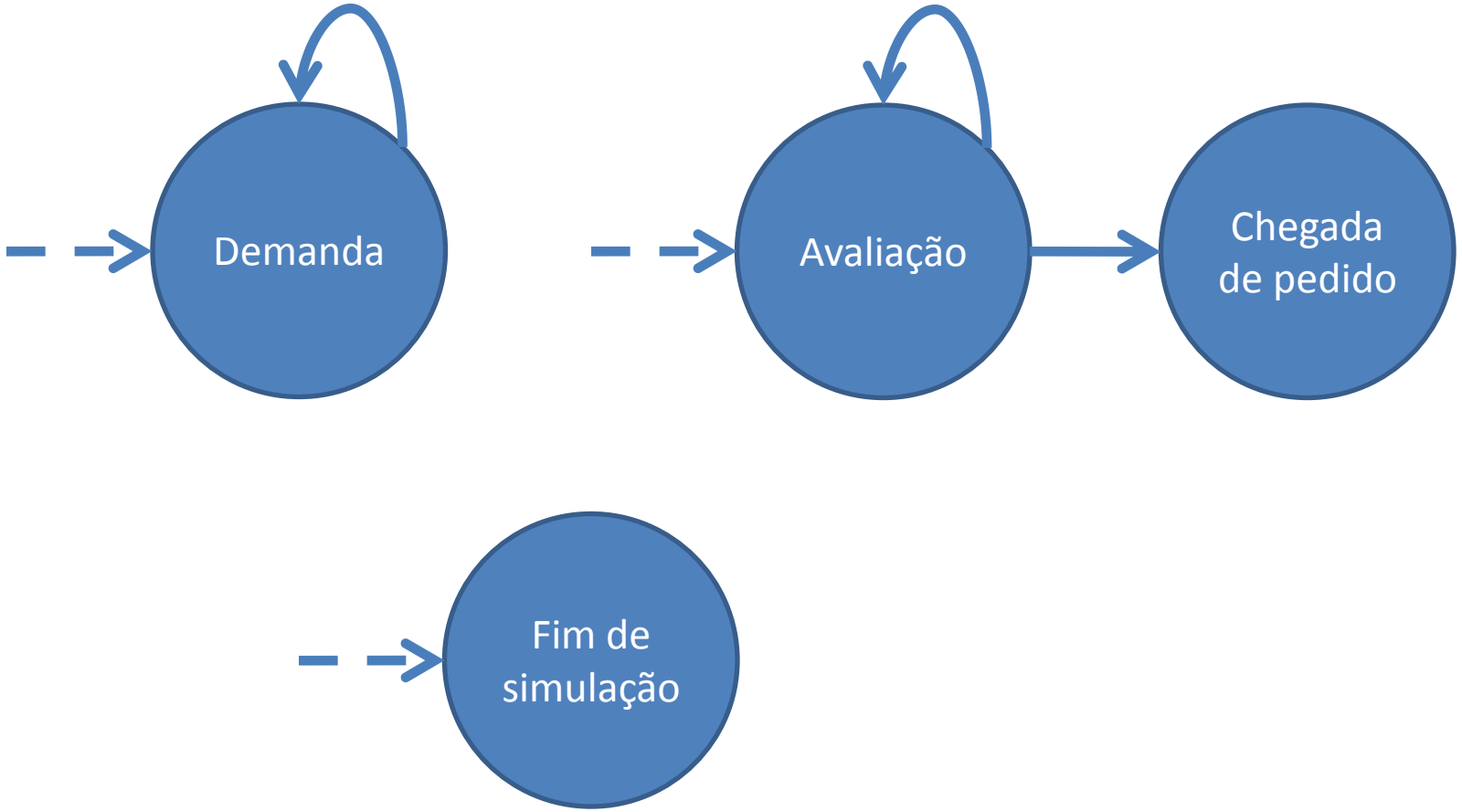
Simulação de sistema de estoque

- Quantos itens comprar por mês (120 meses)?
- Tempo entre demandas: v.a. exponencial i.i.d. com média 0,1 meses
- Demanda: v.a. i.i.d. valendo 1 (p=1/6), 2 (p=1/3), 3 (p=1/3) ou 4 (p=1/6)
- Tempo de chegada do pedido: v.a. uniforme i.i.d. entre 0,5 e 1 mês
- Custo de compra de Z itens = 32 + 3.Z, ou 0 (Z=0)
- Custo de armazenamento = 1 por item por mês
- Custo demanda não atendida = 5 p/ item p/ mês

Simulação de sistema de estoque

- Estoque inicial: 60 itens
- Política de reposição (S,s):
  - Z = S – I, se I < s, ou zero caso contrário
  - Onde I é o nível do estoque
  - Avaliação uma vez por mês
- Alternativas: conjunto definido de pares (S,s)
- Performance: custo total (soma dos custos médios mensais de compra, armazenamento e demanda perdida)

Simulação de sistema de estoque: eventos



Rotina do evento DEMANDA

- Gere o tamanho da demanda
- Decremente o nível do estoque por esta demanda
- Escalone o próximo evento de demanda

Rotina do evento AVALIAÇÃO

- Se o nível do estoque é menor que s:
  - Determine tamanho do pedido (= S – I)
  - Adicione o custo desta compra
  - Escalone o evento de chegada deste pedido
- Escalone o próximo evento de avaliação

Rotina do evento CHEGADA DE PEDIDO

- Incremente o nível do estoque com o tamanho do último pedido
- Indique que não existe próximo evento (tempo infinito)

- Obs.: note que o tempo de entrega do pedido é inferior a 1 mês, e todo mês ocorre avaliação

Rotina de atualização dos acumuladores com médias no tempo

- Executado antes de qualquer rotina de evento

- Se o nível do estoque estava positivo
  - Incremente custo de armazenamento
- Se o nível do estoque estava negativo
  - Incremente custo de demanda não atendida

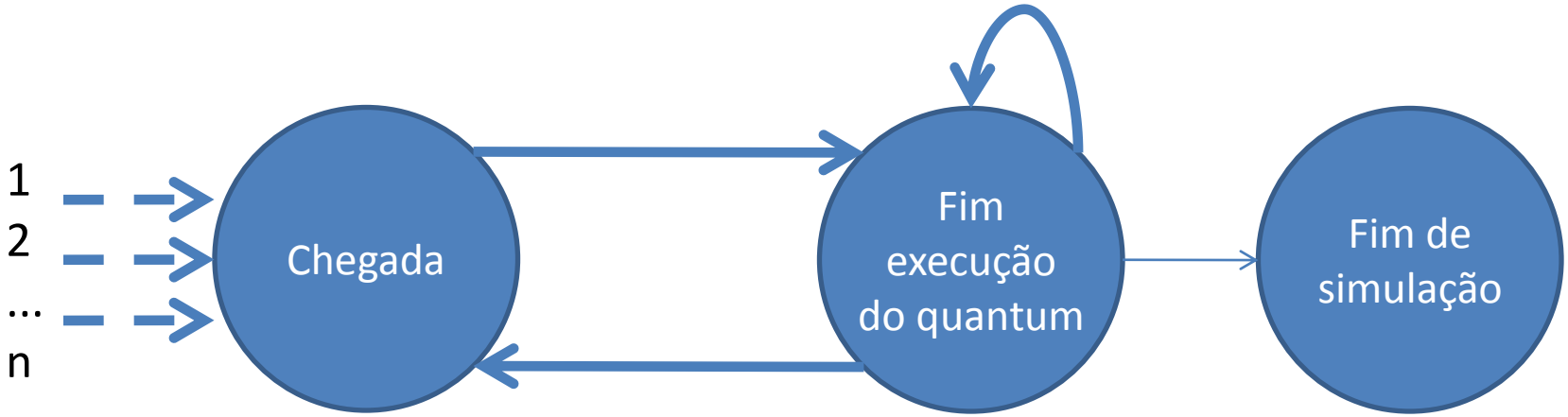
Simulação de computador time-sharing

- Uma CPU e n terminais
- O operador de cada terminal “pensa” por um tempo com dist. exp. com média 25s, e envia um job para a CPU
  - Volta a “pensar” quando seu job termina
- Tempo de execução do job tem dist. exp. com média 0,8s

Simulação de computador time-sharing

- Execução dos jobs pendentes em round-robin:
  - Quantum q=1s por job, se job restante > q
  - Swap time t=0,015s
- Performance:
  - Tempo de resposta médio esperado
  - Média no tempo do número de jobs na fila
  - Utilização esperada da CPU
- Alternativas: n=10,20,...,80
- Quantos terminais de modo que o tempo de resposta seja no máximo 30s ?
- Fim de simulação: 1000 jobs completados

Simulação de computador time-sharing: eventos



Rotina do evento CHEGADA

- Coloque job na fila
- Se a CPU está ociosa,
  - Chame rotina START

Rotina START

- Remova job da fila e calcule tempo de CPU
- Decremente o tempo restante deste job
- Coloque job na CPU
- Escalone evento FIM EXECUÇÃO DO QUANTUM para este job

Rotina do evento FIM EXECUÇÃO DO QUANTUM

- Remova o job da CPU
- Se o job precisa de mais tempo,
  - Coloque o job no final da fila
  - Chame rotina START
- Senão
  - Calcule tempo de resposta
  - Escalone evento CHEGADA para este job
  - Incremente contador de jobs concluídos
  - Se número suficiente de jobs, escalone fim de simulação para este instante
  - Senão, se tem jobs na fila chame rotina START

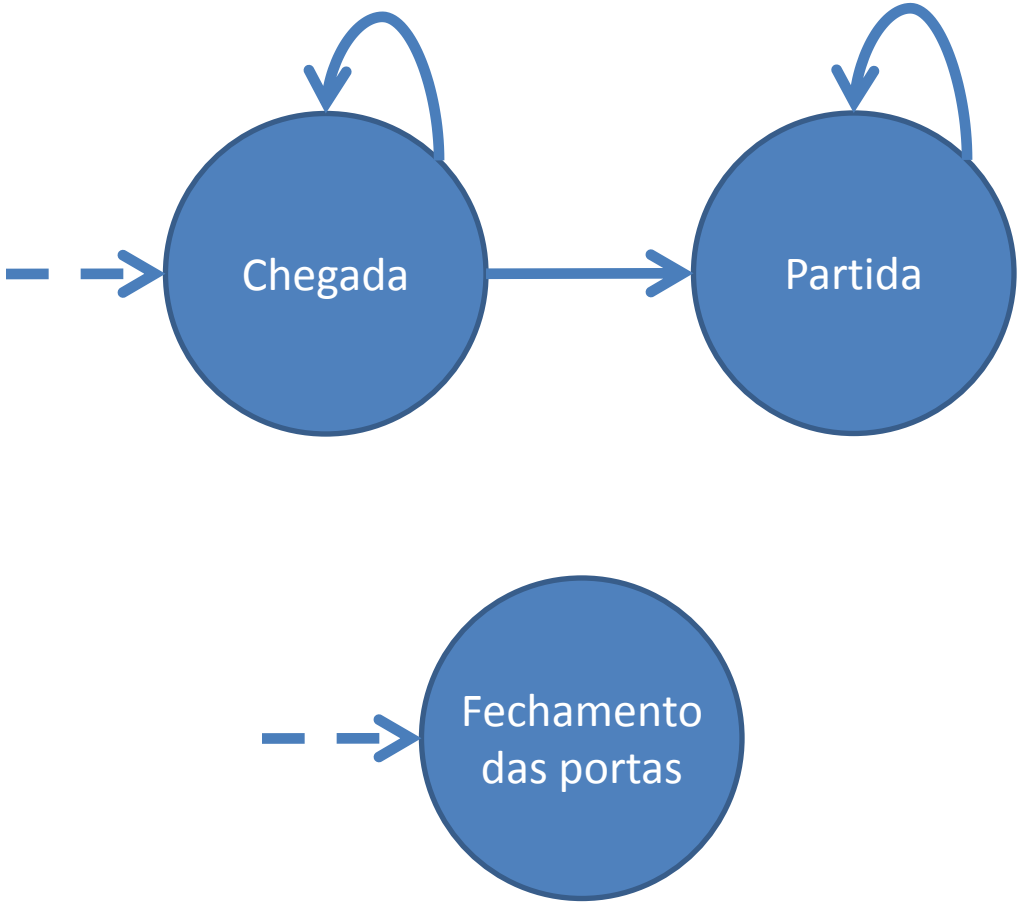
Simulação de banco com múltiplos caixas e troca de fila

- Banco com 5 caixas, abre 9h e fecha 17h
- Opera até que todos os clientes no banco às 17h sejam atendidos
- Tempo entre chegadas: exp. com média 1min
- Tempo atendimento: exp. com média 4,5min
- Cada caixa tem uma fila
- Quem chega escolhe a menor fila
  - Em caso de empate, a mais a esquerda

Simulação de banco com múltiplos caixas e troca de fila

- Quando termina um atendimento, a fila deste caixa x pode ficar 2 clientes menor que de outros (seja S este conjunto destes caixas)
  - Assim, o último cliente (na fila de S que está mais próxima de x) troca para a fila de x
  - Em caso de empate, a fila mais a esquerda
- Alternativas: número de caixas = 4,5,6 e 7
- Performance:
  - Média no tempo do número de clientes na fila
  - Estimativa do tempo médio e máximo na fila

Simulação de banco com múltiplos caixas e troca de fila: eventos



Rotina do evento CHEGADA

- Escalone a próxima chegada
- Se algum caixa está livre,
  - Faça o atraso deste cliente igual a zero
  - Marque este caixa como ocupado
  - Escalone o evento PARTIDA deste cliente
- Senão,
  - Coloque este cliente na menor fila mais a esquerda

Rotina do evento PARTIDA

- Se a fila deste caixa estiver vazia,
  - Marque este caixa como ocioso
- Senão,
  - Retire o primeiro cliente da fila
  - Calcule o tempo na fila deste cliente
  - Escalone o evento PARTIDA deste cliente
- Chame a rotina TROCA_FILA

Rotina TROCA_FILA

- Se existe algum cliente c para trocar de fila,
  - Remova o cliente c do final da fila
  - Se o caixa x que vai receber o cliente está ocupado,
    - Coloque o cliente no final da fila de x
  - Senão,
    - Calcule o tempo em fila do cliente c
    - Marque o caixa como ocupado
    - Escalone o evento PARTIDA do cliente c

Rotina do evento FECHAMENTO DAS PORTAS

- Indique que nenhum outro evento CHEGADA pode ser executado

Simulação de uma fábrica

- Usado para identificar gargalos no processo
  - Rede de filas com múltiplos servidores
- 5 grupos contendo 3,2,4,3 e 1 máquinas identicas
- Tempo entre chegadas: exp. com média 0,25h
- 3 tipos de jobs com probabilidades 0.3,0.5,0.2
- Rotas:
  - Job tipo 1 -> grupos 3,1,2,5
  - Job tipo 2 -> grupos 4,1,3
  - Job tipo 3 -> grupos 2,5,1,4,3

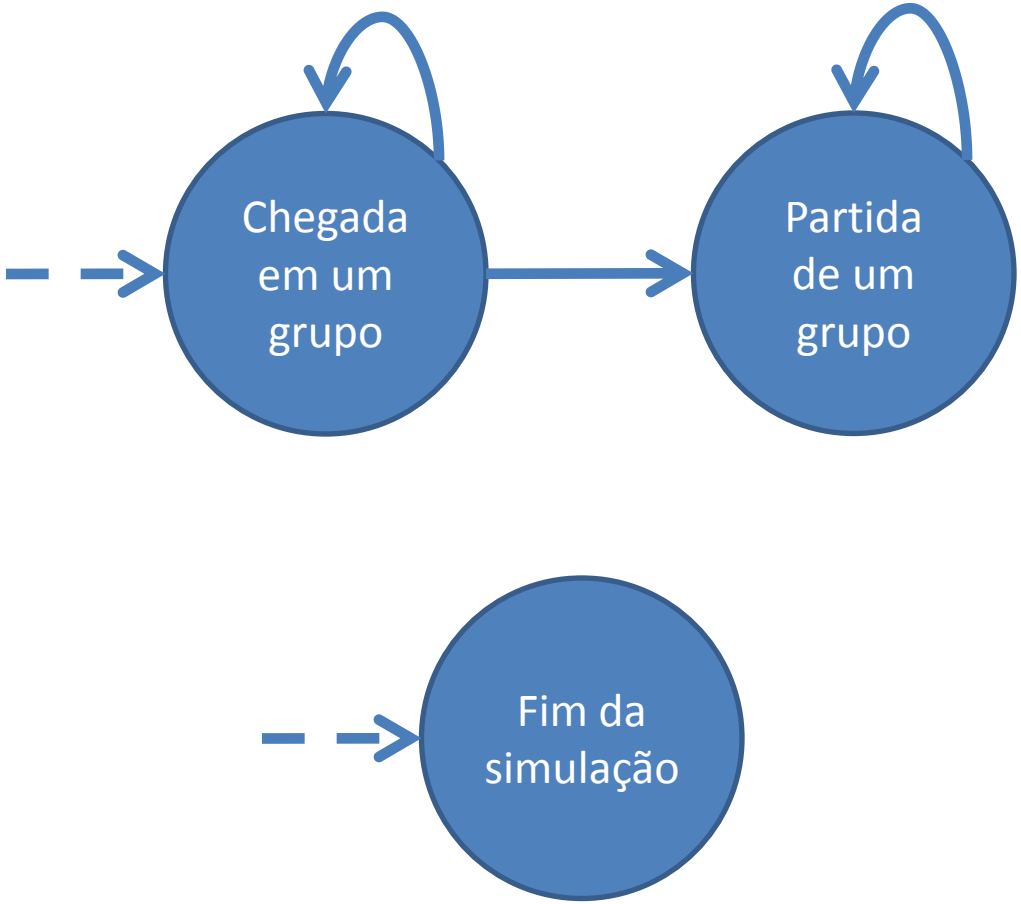
Simulação de uma fábrica

- Se o job encontra todas as máquinas do grupo ocupadas, entra na fila do grupo
- Tempo para executar tarefa: v.a. 2-Erlang com média que depende do job e do grupo
  - Job tipo 1 -> 0.5, 0.6, 0.85, 0.5
  - Job tipo 2 -> 1.1, 0.8, 0.75
  - Job tipo 3 -> 1.2, 0.25, 0.7, 0.9, 1.0
- Simulação de 365 dias, 8h por dia

Simulação de uma fábrica

- Performance:
  - Estimativa do atraso total médio de um job
- Podemos comprar mais uma máquina.
  De qual grupo comprar?
  - Alternativas: estado atual e 1 máquina a mais em cada grupo

Simulação de uma fábrica: eventos



Rotina do evento CHEGADA EM UM GRUPO

- Se for a chegada de um novo job,
  - Escalone a próxima chegada de novo job
  - Gere o tipo deste job e marque TASK=1
- Determine o grupo corrente deste job
- Se todas as mqs do grupo estão ocupadas,
  - Coloque este job no final da fila do grupo
- Senão,
  - Indique atraso zero para este job
  - Marque uma mq ociosa do grupo como ocupada
  - Escalone o evento PARTIDA para este job

Rotina do evento PARTIDA DE UM GRUPO

- Determine o grupo de partida do job
- Se a fila deste grupo está vazia,
  - Marque uma máquina deste grupo como ociosa
- Senão,
  - Remova o primeiro job da fila
  - Calcule o atraso deste job
  - Escalone um evento PARTIDA para este job
- Se o job partindo tem mais alguma tarefa,
  - Incremente a variável TASK deste job
  - Chame a rotina CHEGADA indicando não ser novo job