Implementation / Programming: Procedural Programming Conceptual Frameworks

OSMAN BALCI
Professor

Department of Computer Science
Virginia Polytechnic Institute and State University (Virginia Tech)
Blacksburg, VA 24061, USA

https://manta.cs.vt.edu/balci

Four Conceptual Frameworks for Discrete Simulation Programming in a High-Level Programming Language under the Procedural Paradigm

M&S Areas

A. Based on Model Representation:

1. **Discrete M&S**
2. **Continuous M&S**
3. **Monte Carlo M&S**
4. **System Dynamics M&S**
5. **Gaming-based M&S**
6. **Agent-based M&S**
7. **AI-based M&S**
8. **VR-based M&S**

Discrete M&S Design Approaches:

- **Object-Oriented Design**
- **Procedural Design**

Procedural Programming Conceptual Frameworks:

- **Event Scheduling**
- **Activity Scanning**
- **Three-Phase Approach**
- **Process Interaction**

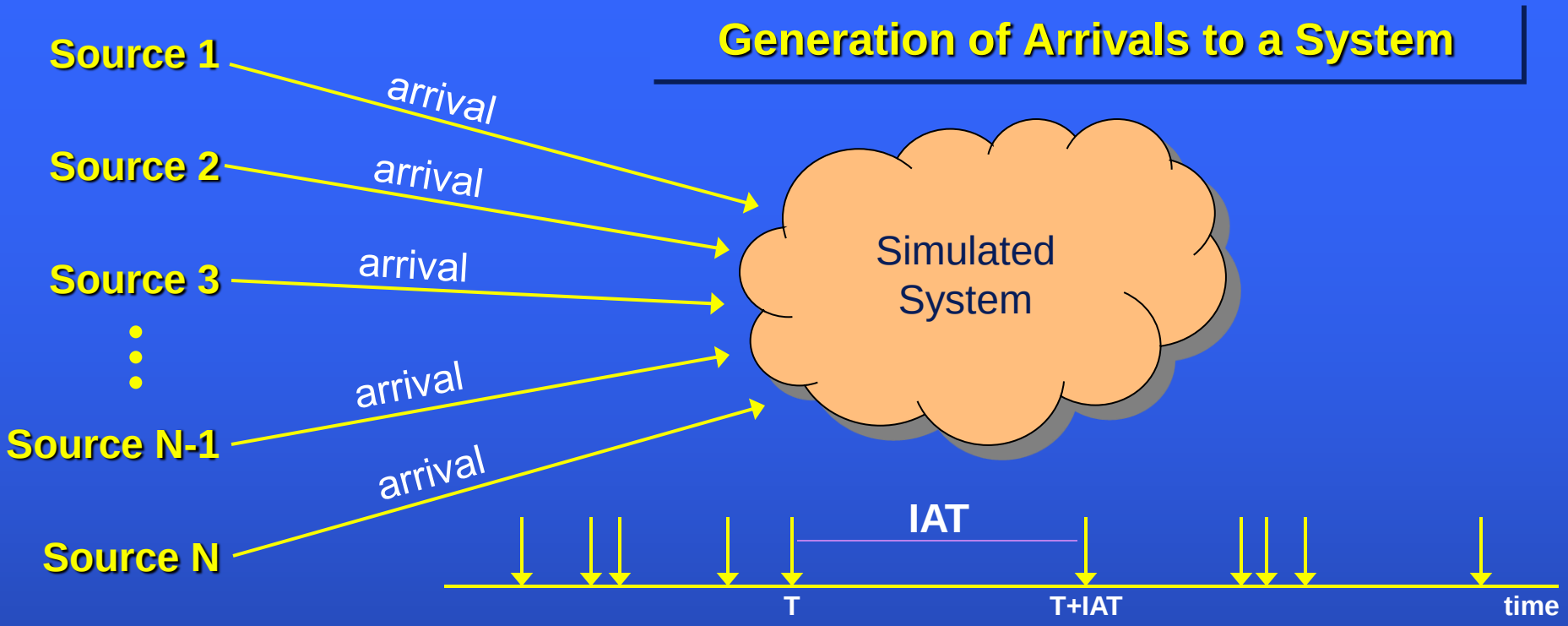
Osman Balci (1988), "The Implementation of Four Conceptual Frameworks for Simulation Modeling in High-Level Languages," In *Proceedings of the 1988 Winter Simulation Conference* (San Diego, CA, Dec. 12-14). IEEE, Piscataway, NJ, pp. 287-295.

Time Flow Mechanism (TFM)

- TFM is that portion of a simulation that
  - Advances time in the simulation, and
  - Provides synchronization of the various parts of the simulation

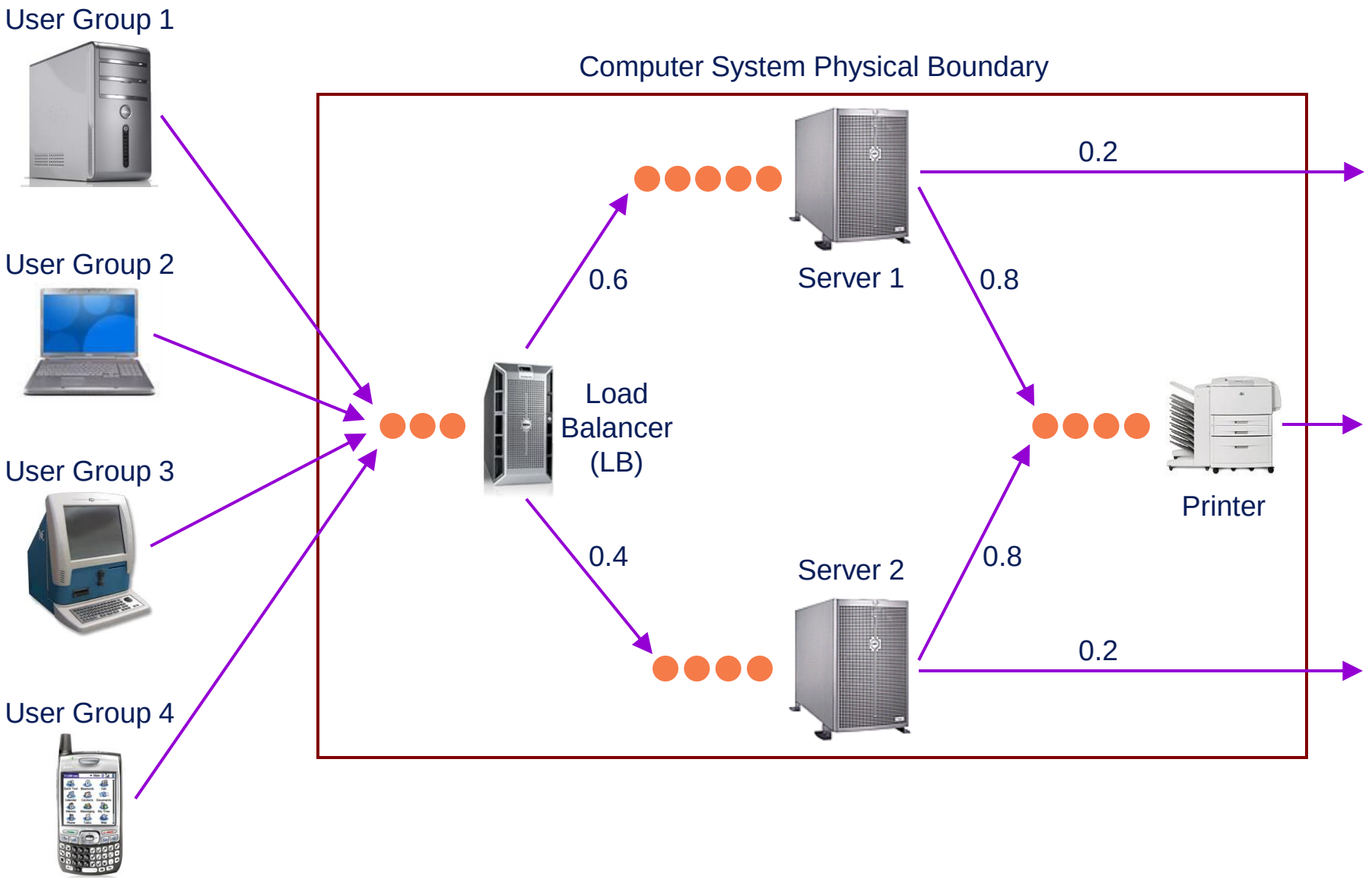
- There are two basic types of TFMs:
  - **Fixed Time Increment TFM**
    The time (simulation clock) is advanced by a small fixed amount.
  - **Variable Time Increment or Next Event TFM**
    The time (simulation clock) is advanced from one event to the next.

Generation of Arrivals to a System



- When an entity such as message, customer, packet, job, signal, etc.) *externally* arrives at time T to the system simulated from source S:
  - Hold the execution of the current arrival at time T temporarily.
  - Generate a random inter-arrival time (IAT) from the probability distribution representing the arrival process from source S.
  - Add the random IAT to the time T, to determine the arrival time of the next entity from that same source.
  - Schedule the arrival of the next entity in the simulation from the same source S at the future arrival time computed as T + IAT
  - Execute the current arrival at time T.

Example System to Simulate



Characterization of Random Variables

User Group	Interarrival Times Probability Distribution	Mean Value (in seconds)
1	Exponential	3200
2	Exponential	640
3	Exponential	1600
4	Exponential	266.67

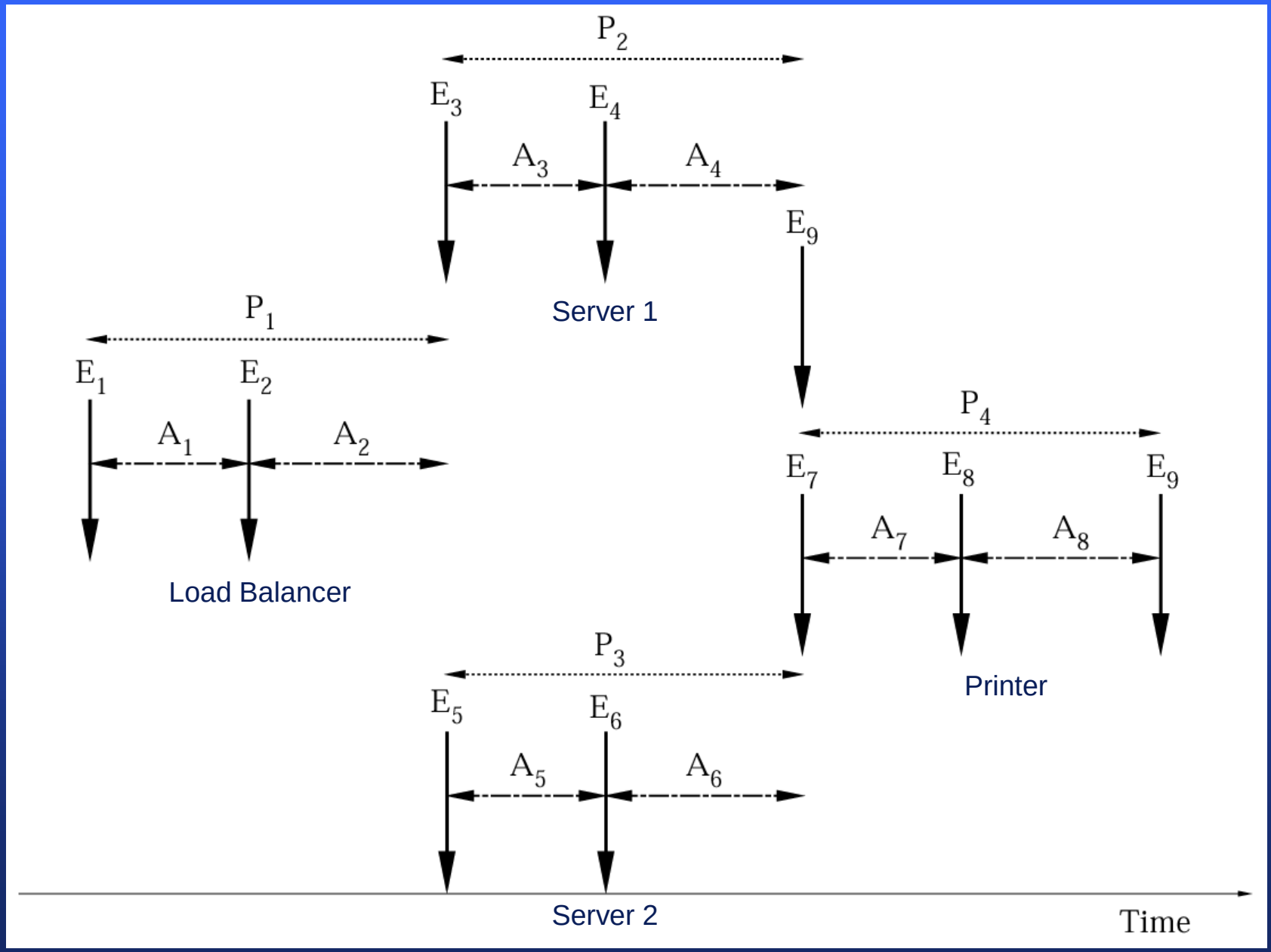
Facility	Processing Times Probability Distribution	Mean Value (in seconds)
Load Balancer	Exponential	112
Server 1	Exponential	226.67
Server 2	Exponential	300
Printer	Exponential	160

Performance Measures of Interest

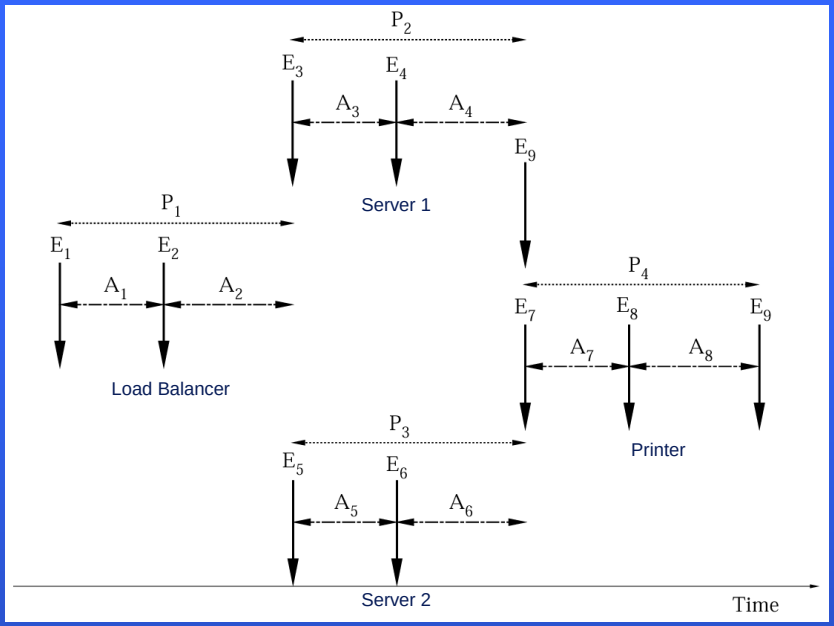
Assuming that the simulation model reaches steady-state conditions after 3,000 jobs, simulate the system for 15,000 jobs in steady state and construct confidence intervals for the following performance measures of interest:

1. Utilization of the Load Balancer (ρ_{LB})
2. Utilization of the Server 1 (ρ_{S1})
3. Utilization of the Server 2 (ρ_{S2})
4. Utilization of the Printer (ρ_{P})
5. Average time spent by a job in the computer system (W)
6. Average number of jobs in the computer system (L)

Events, Activities, and Processes of the Example Problem

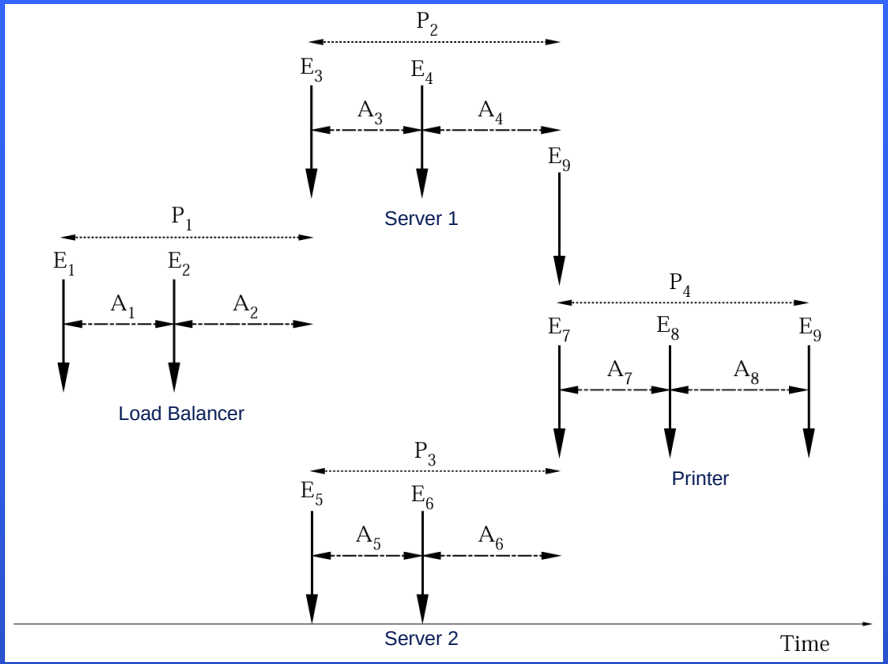


Definition of Events



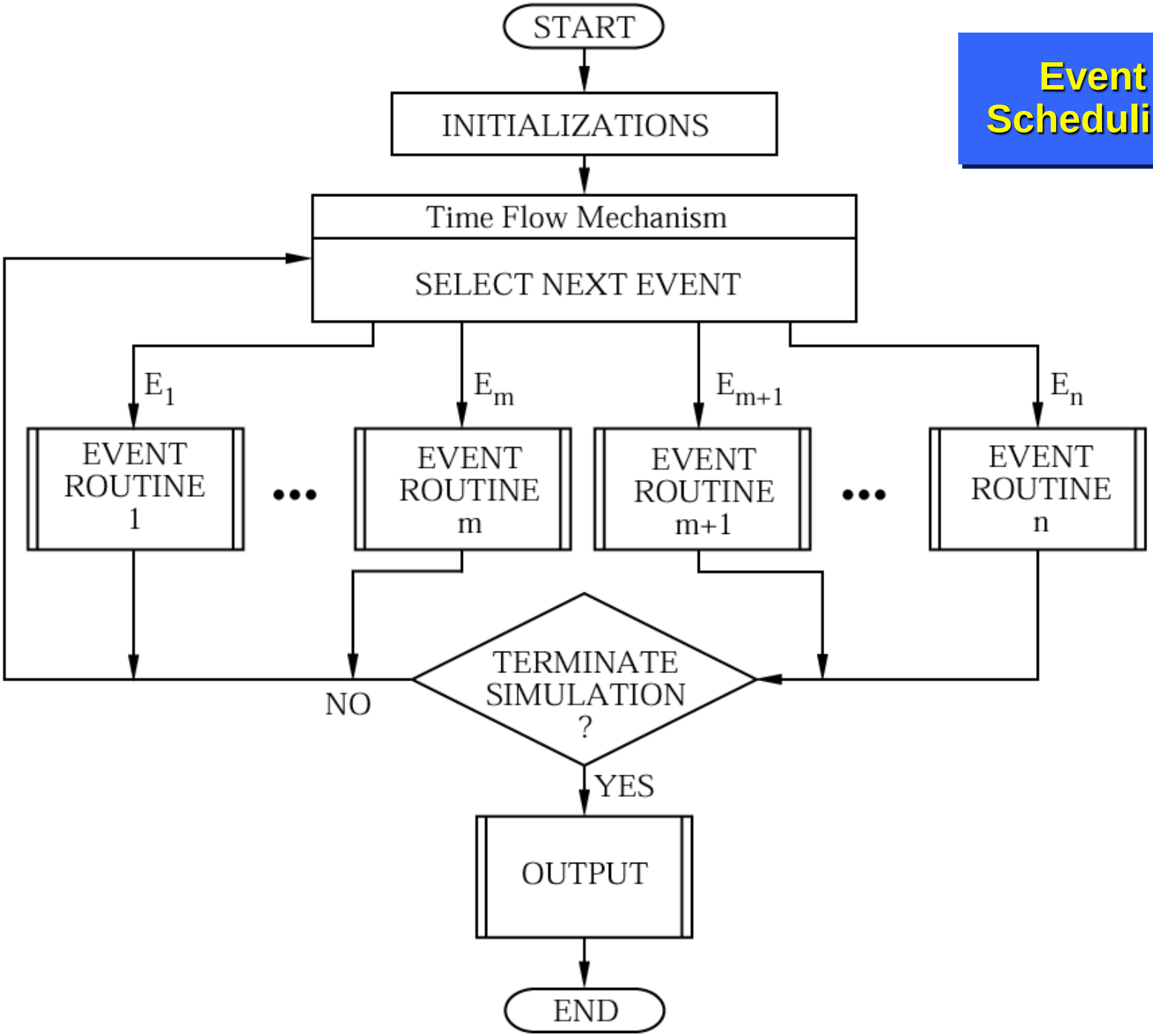
Event	Description	Attribute Value Changed
E_1	Job arrival to the Load Balancer	No. of jobs in system
E_2	Start of Load Balancer processing	Status of Load Balancer
E_3	Job arrival to Server 1	No. of jobs in Server 1
E_4	Start of execution on Server 1	Status of Server 1
E_5	Job arrival to Server 2	No. of jobs in Server 2
E_6	Start of execution on Server 2	Status of Server 2
E_7	Job arrival to the Printer	No. of jobs in Printer
E_8	Start of printing	Status of printer
E_9	Job departure from the computer system	No. of jobs in system

Definition of Activities



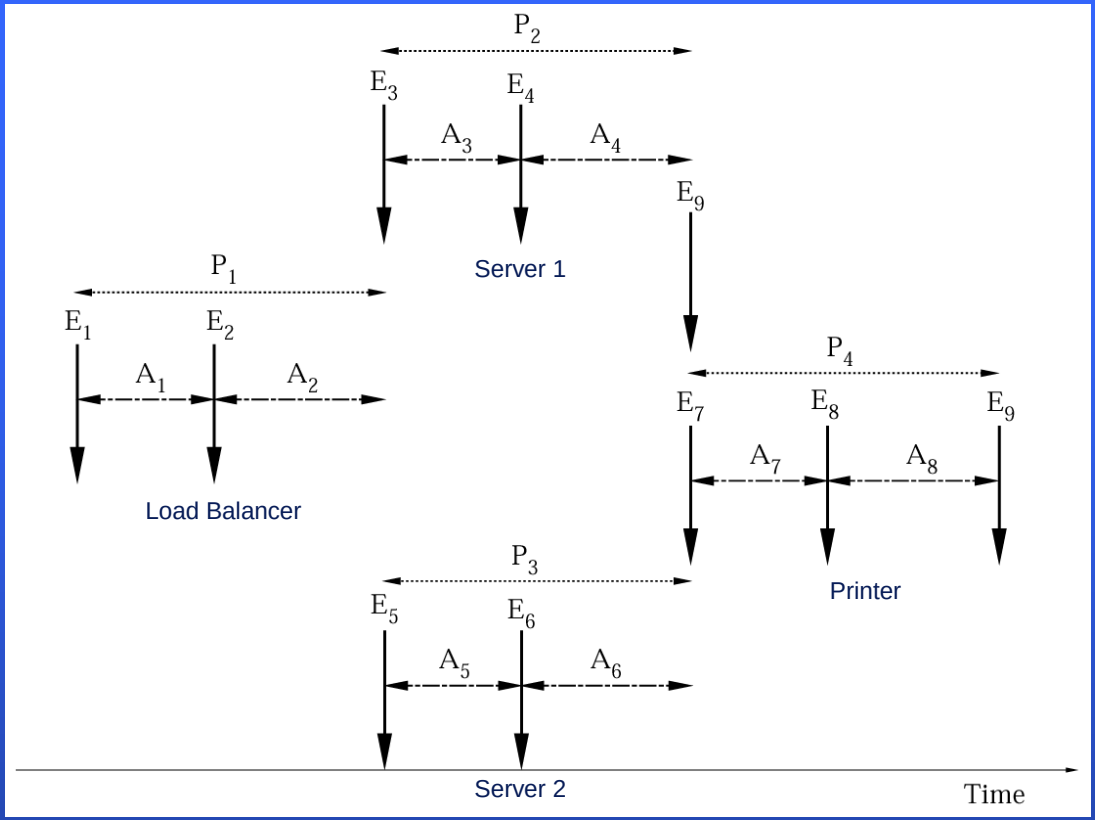
Event	Description	Starting Event	Ending Event
A_1	Waiting for the Load Balancer	E_1	E_2
A_2	Execution on the Load Balancer	E_2	E_3 or E_5
A_3	Waiting for the Server 1	E_3	E_4
A_4	Execution on the Server 1	E_4	E_7 or E_9
A_5	Waiting for the Server 2	E_5	E_6
A_6	Execution on the Server 2	E_6	E_7 or E_9
A_7	Waiting for the Printer	E_7	E_8
A_8	Execution on the Printer	E_8	E_9

Event Scheduling



An Event List

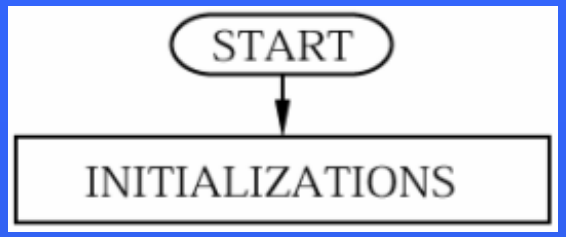
	Attribute 1 **Event Occurrence Time**	*Attribute 2* **Event Identification**	*Attribute 3*	. . .	*Attribute k*
Record 1				. . .	
Record 2				. . .	
Record 3				. . .	
	⋮	⋮	⋮	⋮	⋮
Record r				. . .	



Initializations

- **Create a variable to show when in the future a server will be available so that we can schedule a service in the future:**
  - **timeAtWhichLBWillBeIdle = 0**
  - **timeAtWhichServer1WillBeIdle = 0**
  - **timeAtWhichServer2WillBeIdle = 0**
  - **timeAtWhichPrinterWillBeIdle = 0**

Initializations



Simulation Clock = 0

- As part of Initializations, generate and schedule the arrival of the first job from each of the four user groups.
- The Interarrival Time random variable has an Exponential probability distribution with different mean values as given below:

User Group	Interarrival Times Probability Distribution	Mean Value (in seconds)
1	Exponential	3200
2	Exponential	640
3	Exponential	1600
4	Exponential	266.67

- Call the following Exponential RVG program to generate a random value for the Interarrival Time random variable:

```
double ExponentialRVG( double Mean ) {
    return ( – Mean * log( RandomNumberGenerator( &RandomNumberSeed )) );
}
```

Initializations

Simulation Clock = 0

User Group	Interarrival Times Probability Distribution	Mean Value (in seconds)
1	Exponential	3200
2	Exponential	640
3	Exponential	1600
4	Exponential	266.67

- Assume that calling ExponentialRVG(3200) returns 2567 as a random value for the Interarrival time. Add it to the current simulation clock value 0 to obtain an arrival time of 2567.
- Create an event record as follows:

Attribute 1 Event Occurrence Time	Attribute 2 Event ID	Attribute 3 Arrival Source	Attribute 4 Arrival Time
2567	E_1	UG1	2567

- Insert this record in the Event List with respect to the Event Occurrence Time.

Initializations

Simulation Clock = 0

User Group	Interarrival Times Probability Distribution	Mean Value (in seconds)
1	Exponential	3200
2	Exponential	640
3	Exponential	1600
4	Exponential	266.67

- Assume that calling ExponentialRVG(640) returns 434 as a random value for the Interarrival time. Add it to the current simulation clock value 0 to obtain an arrival time of 434.
- Create an event record as follows:

Attribute 1 Event Occurrence Time	Attribute 2 Event ID	Attribute 3 Arrival Source	Attribute 4 Arrival Time
434	E_1	UG2	434

- Insert this record in the Event List with respect to the Event Occurrence Time.

Initializations

Simulation Clock = 0

User Group	Interarrival Times Probability Distribution	Mean Value (in seconds)
1	Exponential	3200
2	Exponential	640
3	Exponential	1600
4	Exponential	266.67

- Assume that calling ExponentialRVG(1600) returns 1800 as a random value for the Interarrival time. Add it to the current simulation clock value 0 to obtain an arrival time of 1800.
- Create an event record as follows:

Attribute 1 Event Occurrence Time	Attribute 2 Event ID	Attribute 3 Arrival Source	Attribute 4 Arrival Time
1800	E_1	UG3	1800

- Insert this record in the Event List with respect to the Event Occurrence Time.

Initializations

Simulation Clock = 0

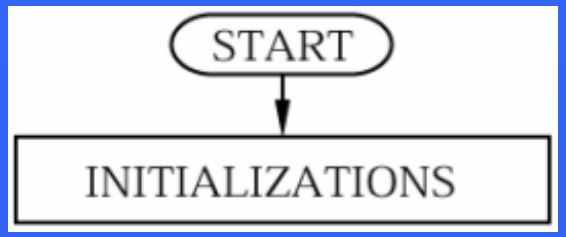
User Group	Interarrival Times Probability Distribution	Mean Value (in seconds)
1	Exponential	3200
2	Exponential	640
3	Exponential	1600
4	Exponential	266.67

- Assume that calling ExponentialRVG(266.67) returns 305 as a random value for the Interarrival time. Add it to the current simulation clock value 0 to obtain an arrival time of 305.
- Create an event record as follows:

Attribute 1 Event Occurrence Time	Attribute 2 Event ID	Attribute 3 Arrival Source	Attribute 4 Arrival Time
305	E_1	UG4	305

- Insert this record in the Event List with respect to the Event Occurrence Time.

Initializations



Simulation Clock = 0

User Group	Interarrival Times Probability Distribution	Mean Value (in seconds)
1	Exponential	3200
2	Exponential	640
3	Exponential	1600
4	Exponential	266.67

Random Variates

2567
434
1800
305

The Event List after the Initializations:

Attribute 1 Event Occurrence Time	Attribute 2 Event ID	Attribute 3 Arrival Source	Attribute 4 Arrival Time
305	E_1	UG4	305
434	E_1	UG2	434
1800	E_1	UG3	1800
2567	E_1	UG1	2567

Time Flow Mechanism

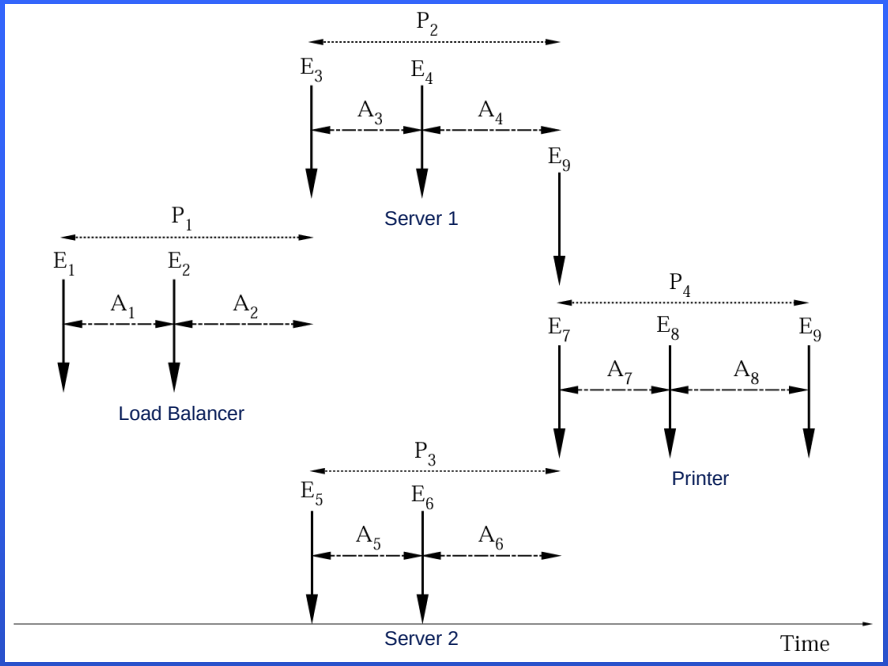
Time Flow Mechanism
SELECT NEXT EVENT

Next event to select is always the event on the top of the Event List.

Therefore, the simulation clock value is advanced from the current value to the Event Occurrence Time of the Event on the top of the Event List.

Simulation Clock = 0 ⇒ Simulation Clock = 305

Event Occurrence Time	Event ID	Arrival Source	Arrival Time
305	E_1	UG4	305
434	E_1	UG2	434
1800	E_1	UG3	1800
2567	E_1	UG1	2567



Execute Event E_1 at Time 305

Simulation Clock = 305

Event Occurrence Time	Event ID	Arrival Source	Arrival Time
305	E_1	UG4	305

E1 Event Execution Program:

- Since the E_1 event designates an arrival of a new job to the system, before we execute it, we first generate and schedule the arrival of the next job from the same source.
- To do this, see the next slide.

Scheduling the Arrival of the Next Job

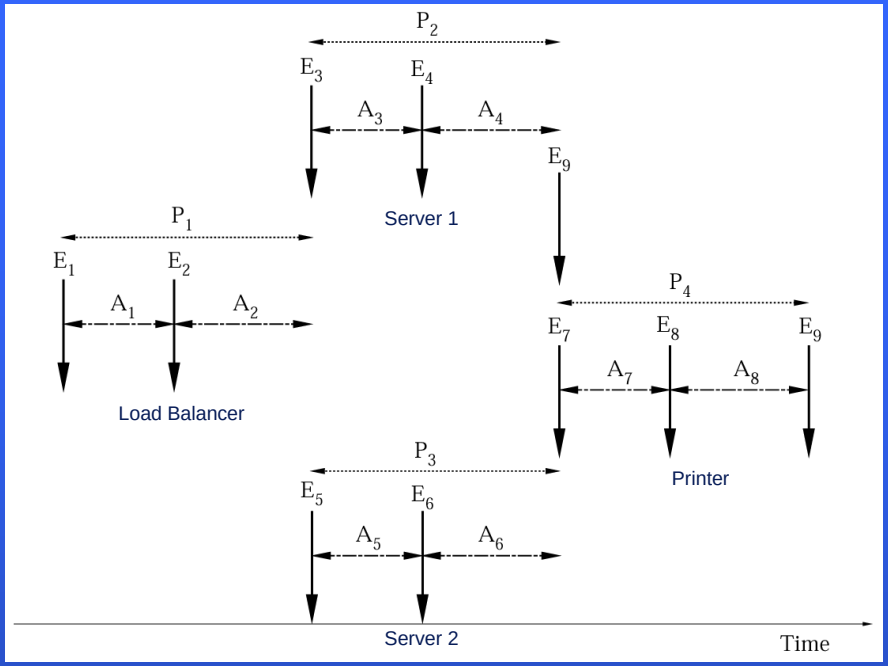
Simulation Clock = 305

User Group	Interarrival Times Probability Distribution	Mean Value (in seconds)
1	Exponential	3200
2	Exponential	640
3	Exponential	1600
4	Exponential	266.67

- Assume that calling ExponentialRVG(266.67) returns 1645 as a random value for the Interarrival time. Add it to the current simulation clock value 305 to obtain an arrival time of 1950.
- Create an event record as follows:

Attribute 1 Event Occurrence Time	Attribute 2 Event ID	Attribute 3 Arrival Source	Attribute 4 Arrival Time
1950	E_1	UG4	1950

- Insert this record in the Event List with respect to the Event Occurrence Time.



Event E_1 Execution

Simulation Clock = 305

Event Occurrence Time	Event ID	Arrival Source	Arrival Time
305	E_1	UG4	305

timeAtWhichLBWillBeIdle = 0

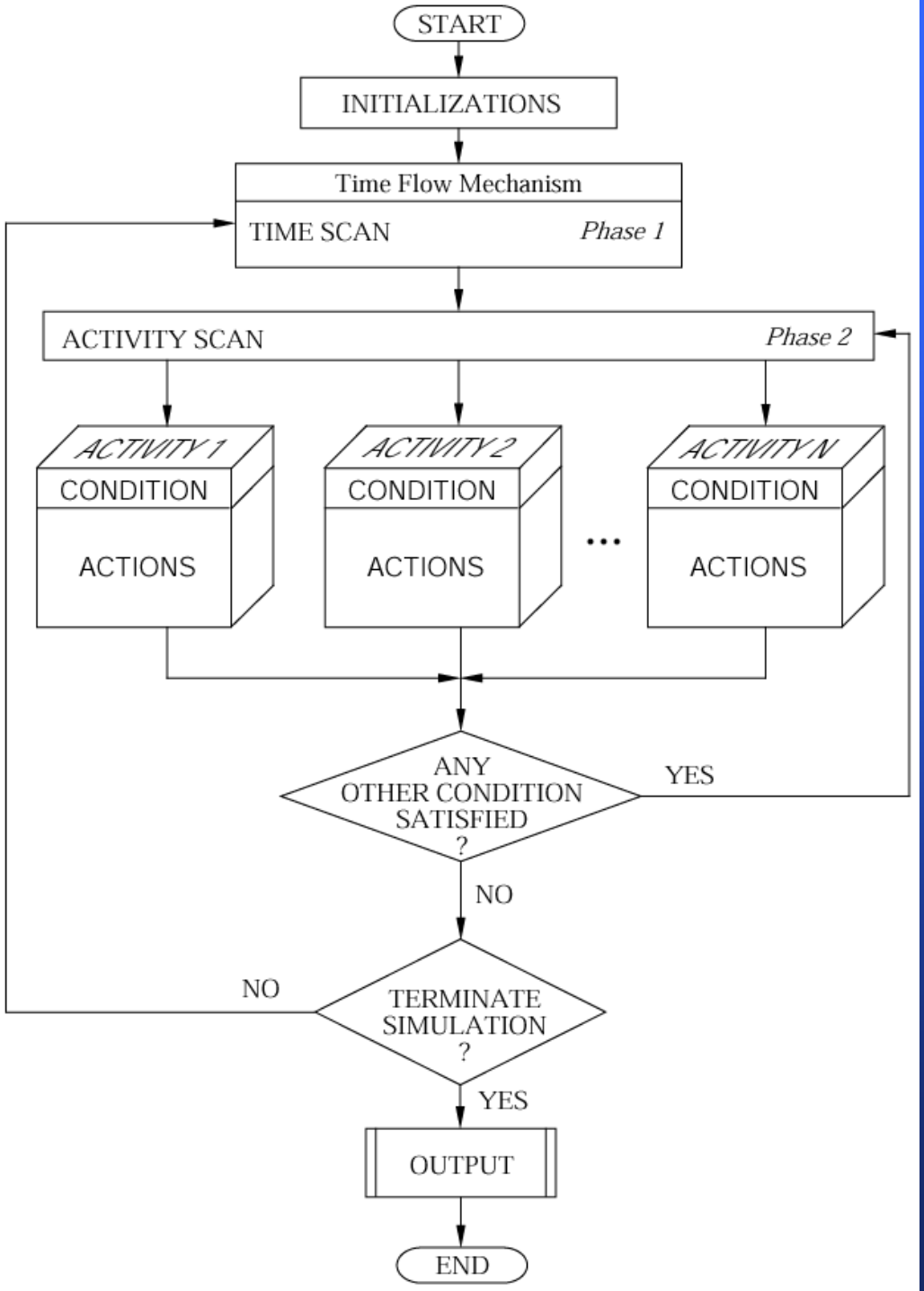
- Since the Load Balancer is idle, the job can enter into service at time 305.
- Assume that calling ExponentialRVG(112) returns 200 as a random value for the service time by the Load Balancer.
- Thus, the arriving job will leave the Load Balancer at time 305+200 = 505
- Set timeAtWhichLBWillBeIdle = 505
- Generate a random number between 0 and 1 to represent the probability of selecting a Server. Assume that it is 0.5 implying the selection of Server1.
- Create a new Event E_3 with occurrence time 505, arrival source UG4, and arrival time 305.
- Insert the new Event E_3 record into the Event List.
- Delete the top Event E_1 just executed from the Event List.
- The new Event List looks like as shown on the next slide.

Event List After the E_1 Event Execution

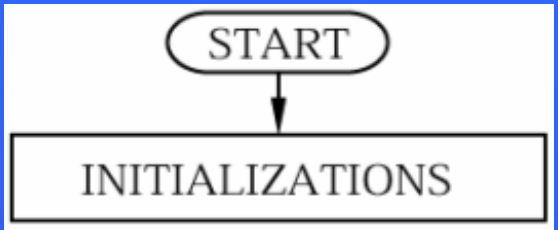
Event Occurrence Time	Event ID	Arrival Source	Arrival Time
~~305~~	~~E_1~~	~~UG4~~	~~305~~
434	E_1	UG2	434
→ 505	E_3	UG4	305
1800	E_1	UG3	1800
→ 1950	E_1	UG4	1950
2567	E_1	UG1	2567

- Go to the Time Flow Mechanism.
- Select the Next Event, which is the event on top of the Event List above.
- Advance the simulation clock to 434. Execute Event E_1 at 434.
- Continue similarly until the simulation is terminated.

Activity Scanning



Initializations



Simulation Clock = 0

User Group	Interarrival Times Probability Distribution	Mean Value (in seconds)
1	Exponential	3200
2	Exponential	640
3	Exponential	1600
4	Exponential	266.67

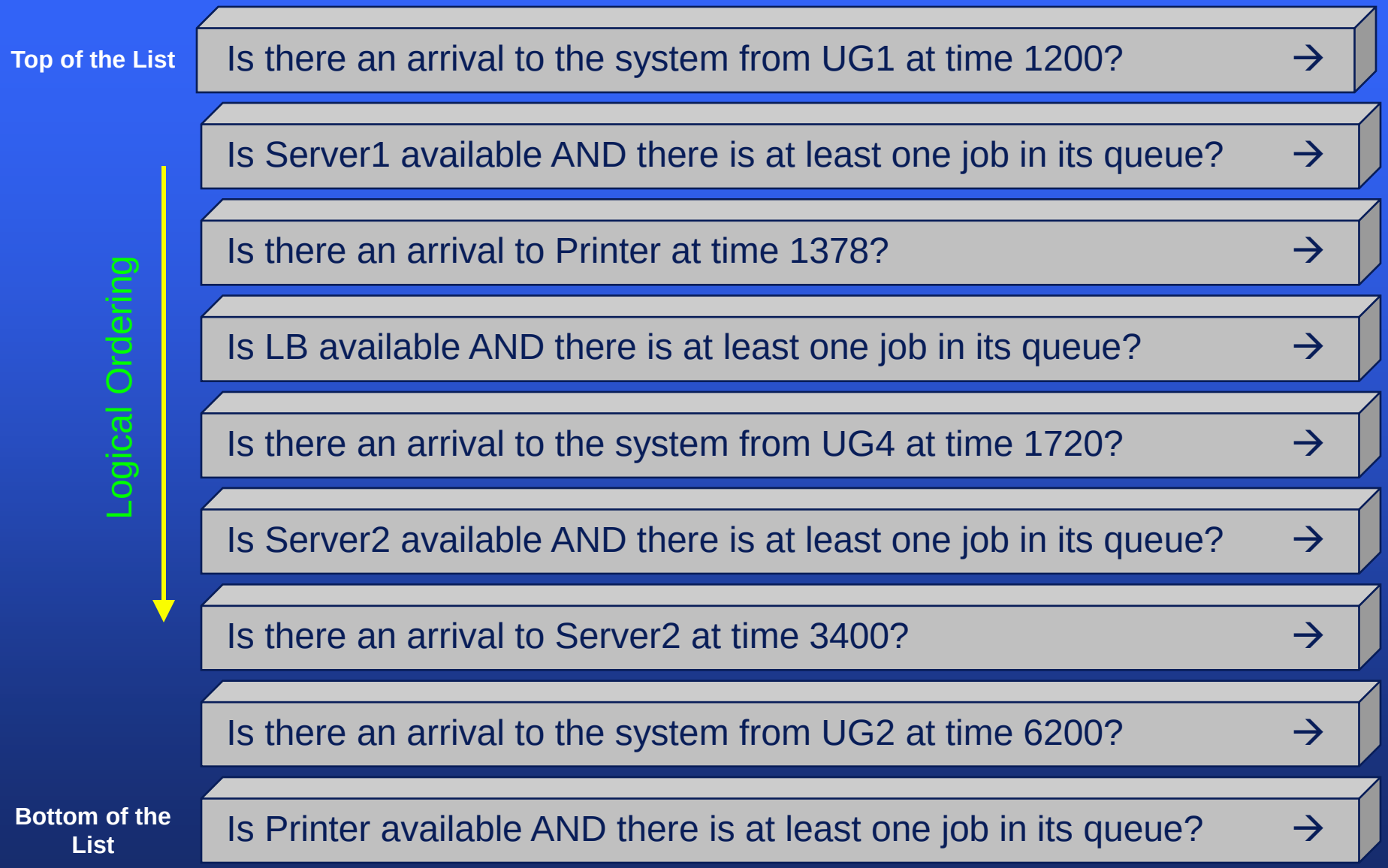
Random Variates

→ 2567
→ 434
→ 1800
→ 305

The Activity (Condition) List after the Initializations:

Is there an arrival from UG4 at time 305?	Is there an arrival from UG2 at time 434?	Is there an arrival from UG3 at time 1800?	Is there an arrival from UG1 at time 2567?
Actions	Actions	Actions	Actions

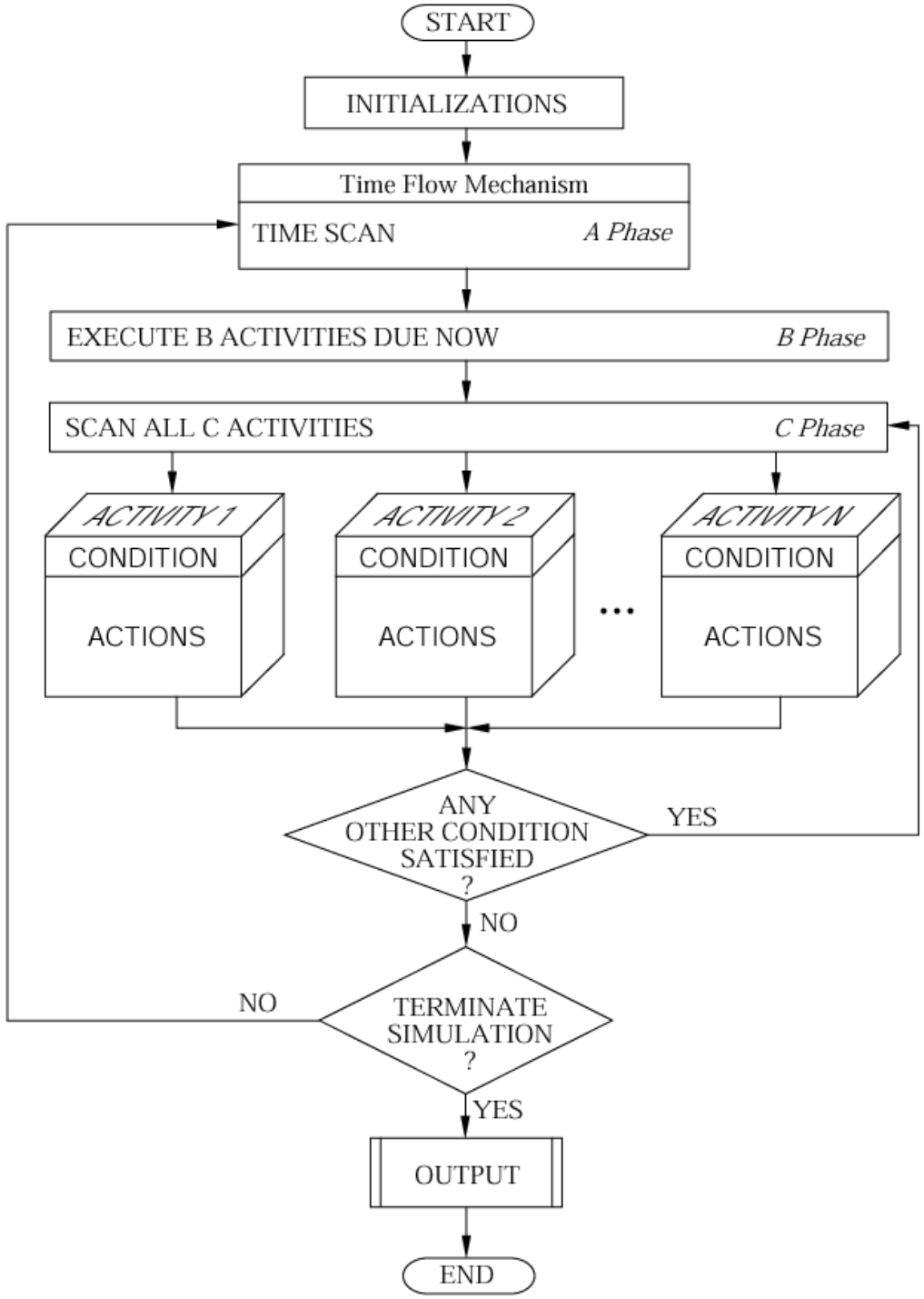
Example Condition List



→ Designates the pointer to the procedure (actions) to be executed if the condition is true.

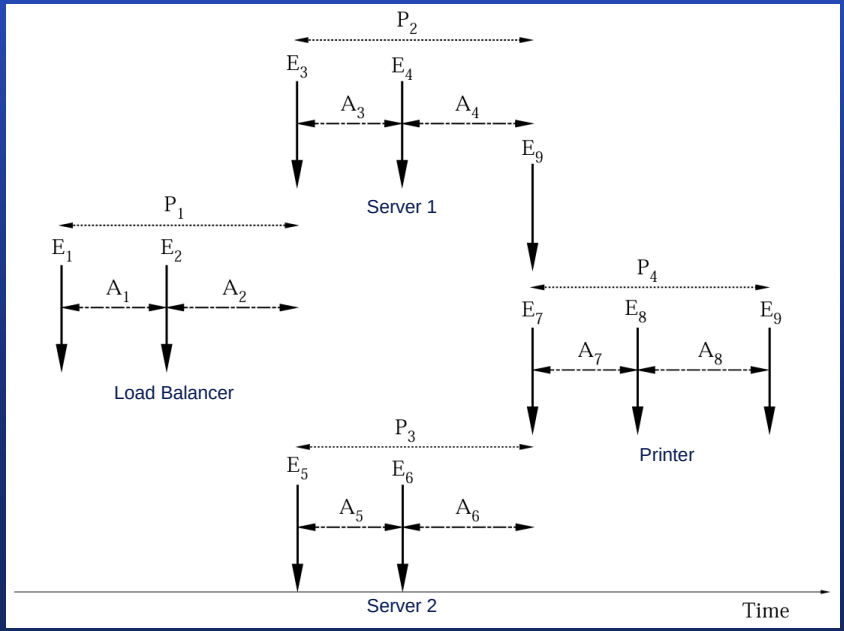
Time Flow Mechanism and Execution Logic

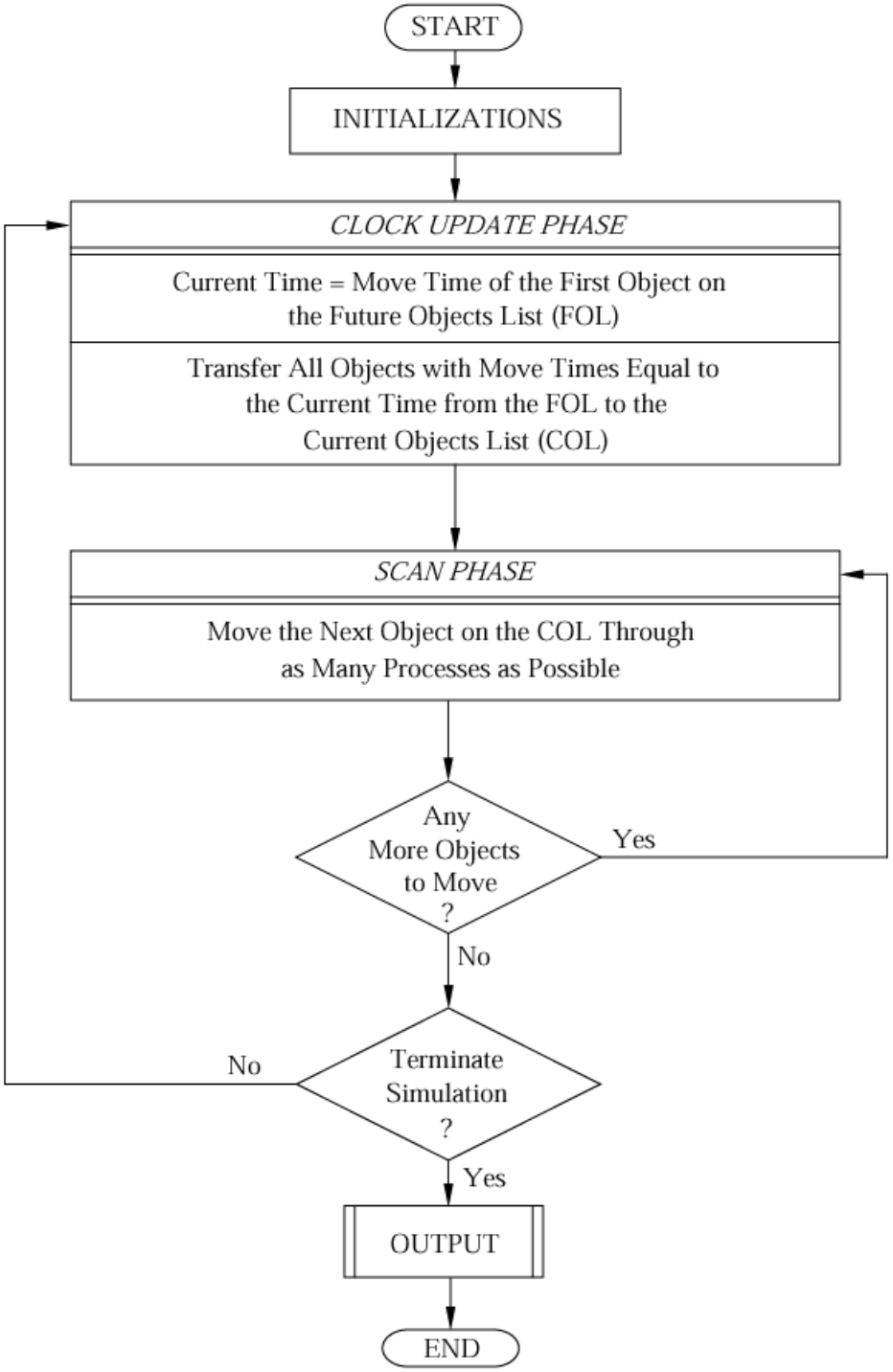
- The Activity Scanning approach is a pure state-based approach for which Fixed Time Increment Time Flow Mechanism (TFM) is used.
- The simulation time (clock) is advanced by a small delta *t* fixed amount. Assume that delta *t* is 1 second for our example problem.
- Increment the simulation clock from 0 to 1 second.
- Scan the activity (condition) list from top to bottom.
- If a condition is satisfied, then execute the corresponding actions. Note that execution of the actions may create new activities to be inserted into the activity list in a logical order.
- If no condition is satisfied, after repeated scan of the condition list, then go to the TFM and increment the simulation clock by delta *t*.
- Execute the activity scan again.
- Continue similarly until the simulation is terminated.



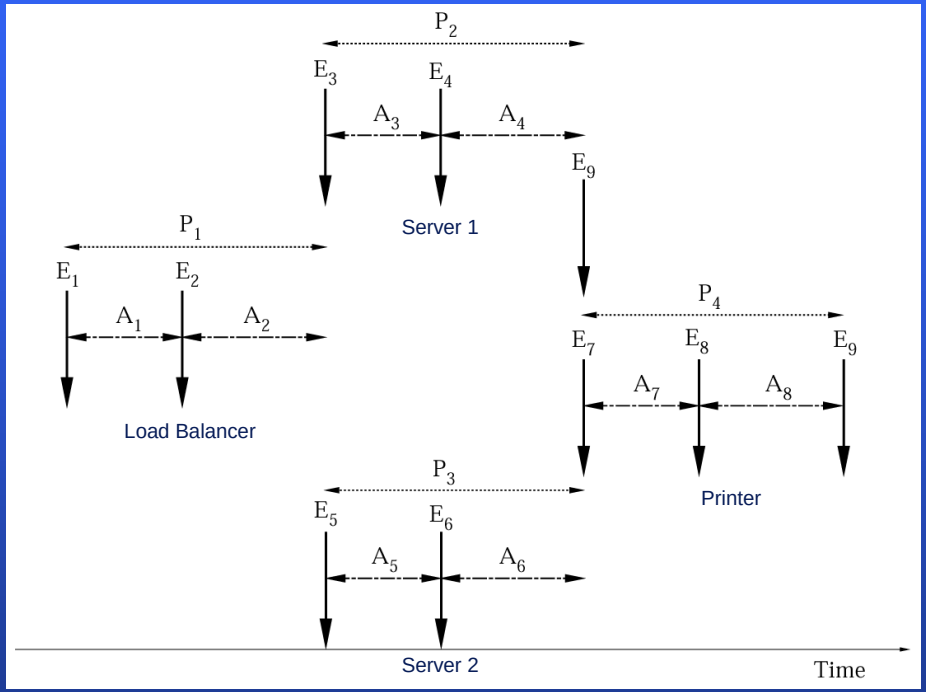
Three Phase Approach

- **B Activities:** are the bound to occur or book-keeping activities
  - e.g., E_k, k = 1,3,5,7,9
- **C Activities:** are the conditional or co-operative activities
  - e.g., A_k, k = 2,4,6,8

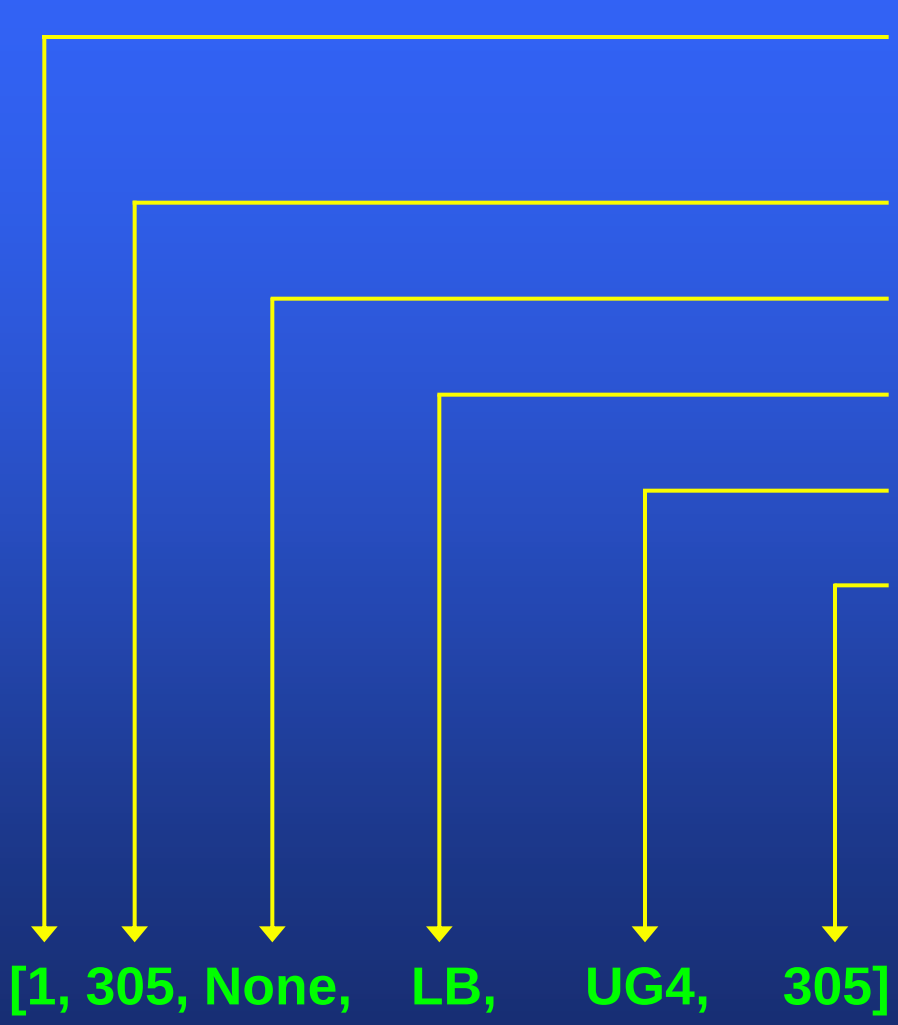




Process Interaction



Attributes for Describing an Object



- Object Identification (ID) Unique Number
- Move Time
- ID of Current Location
- ID of Next Location to Move to
- ID of Source of Arrival
- Arrival Time to System
- Priority Level (if used/needed)
- Other attributes as needed …

"None" implies that the object is outside of the model and it is on its way to the model.

Initializations

Simulation Clock = 0

User Group	Interarrival Times Probability Distribution	Mean Value (in seconds)
1	Exponential	3200
2	Exponential	640
3	Exponential	1600
4	Exponential	266.67

Random Variates

2567 → 1 2567

434 → 2 434

1800 → 3 1800

305 → 4 305

The Future Objects List after the Initializations:

Front

[4, 305, None, LB, UG4, 305]

[2, 434, None, LB, UG2, 434]

[3, 1800, None, LB, UG3, 1800]

[1, 2567, None, LB, UG1, 2567]

Object ID numbers are reused after an object leaves the model or is terminated.

Clock Update Phase

Future Objects List (ordered with respect to move times)

- **Advance the simulation time (clock) to the Move Time of the First Object on the Future Objects List.**

Simulation Clock = 0 ⟹ **Simulation Clock = 305**

- **Transfer all objects with Move Time equal to the current time from the Future Objects List to the Current Objects List.**

Current Objects List (ordered as First Come First Served or any other queue discipline)

Scan Phase

Simulation Clock = 305

Current Objects List

[4, ASAP, None, LB, UG4, 305]

- **Scan Phase: Move the next object on the Current Objects List through as many processes as possible. If the object cannot move to the next location, it stays on the Current Objects List.**
- **Moving Object 4 from NONE to Load Balancer (LB) implies that it is a new arriving job in the model.**
- **Since the object movement designates an arrival of a new job to the system / model, before we execute it, we first generate and schedule the arrival of the next job from the same source.**
- **To do this, see the next slide.**

Scheduling the Arrival of the Next Job

Simulation Clock = 305

User Group	Interarrival Times Probability Distribution	Mean Value (in seconds)
1	Exponential	3200
2	Exponential	640
3	Exponential	1600
4	Exponential	266.67

- Assume that calling ExponentialRVG(266.67) returns 1645 as a random value for the Interarrival time. Add it to the current simulation clock value 305 to obtain an arrival time of 1950.
- Create a new job as follows:

  [5, 1950, None, LB, UG4, 1950]

- Insert this new job in the Future Objects List with respect to its Move Time.

Scan Phase

Simulation Clock = 305

Current Objects List

[4, ASAP, None, LB, UG4, 305]

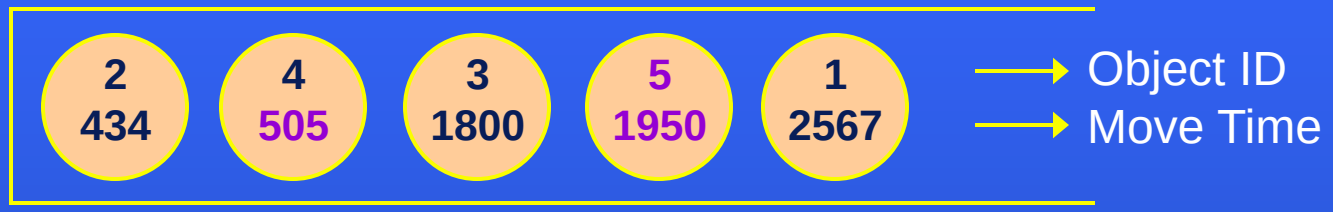
- Since the Load Balancer is idle, the job can enter into service at time 305.
- Assume that calling ExponentialRVG(112) returns 200 as a random value for the service time by the Load Balancer.
- Thus, the arriving job will leave the Load Balancer at time 305+200 = 505
- Set timeAtWhichLBWillBeIdle = 505
- Generate a random number between 0 and 1 to represent the probability of selecting a Server. Assume that it is 0.5 implying the selection of Server1.
- Create a new job as follows:

[4, 505, LB, Server1, UG4, 305]

- Insert the new job into the Future Objects List with respect to its move time.
- The new Future Objects List looks like as shown on the next slide.

After the First Scan Phase

Future Objects List (ordered with respect to move times)



- **Object 4 has a new move time to move from LB to Server1.**
- **Object 5 is a new arriving job inserted.**

Current Objects List

Empty

- **Go to the Clock Update Phase.**
- **Continue executing Clock Update and Scan phases until the simulation is terminated.**