Analysis of System Performance

IN2072

Chapter 0 – Introduction to Modeling

Dr. Alexander Klein
Prof. Dr.-Ing. Georg Carle

Chair for Network Architectures and Services
Department of Computer Science
Technische Universität München
http://www.net.in.tum.de

Topics

- Introduction of basic terms
  - Model, systems, simulation…
- Evaluation spectrum
- When to use simulation
- Typical use cases for simulations

Motivation

- Evaluate Properties of Computer Systems:
  - Correctness (verification)
  - Performance
- Modern Computer Systems:
  - Guarantee minimum performance (e.g. delay for real time systems)
  - Comparison of different systems
- Evaluation:
  - Measurement
  - Modeling
  - Performance Evaluation
- Methods to improve the performance:
  - Design
  - Development
  - Tuning
  - Comparison of different systems

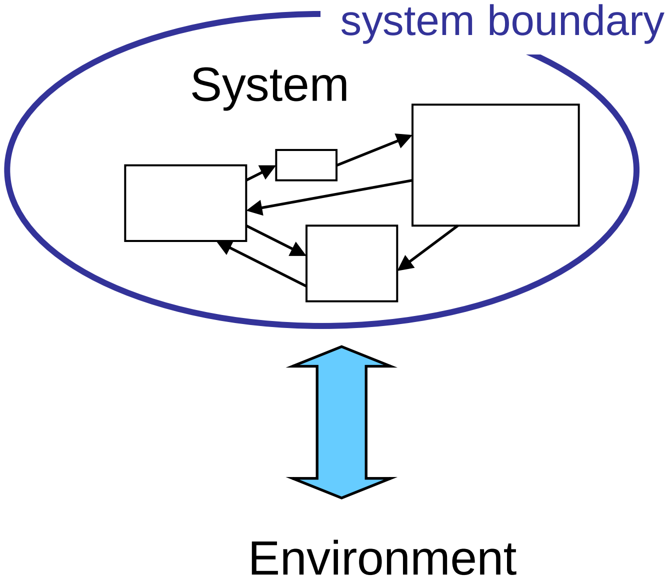
Modeling and Simulation

- **Model:** A representation of a system (or: entity, process, …)
- **Simulation:** The process of exercising a model to characterize the behaviour of the modelled system / entity / process over time
- **Computer simulation:** A simulation where the system doing the emulation is a computer program

What is a system?

System:

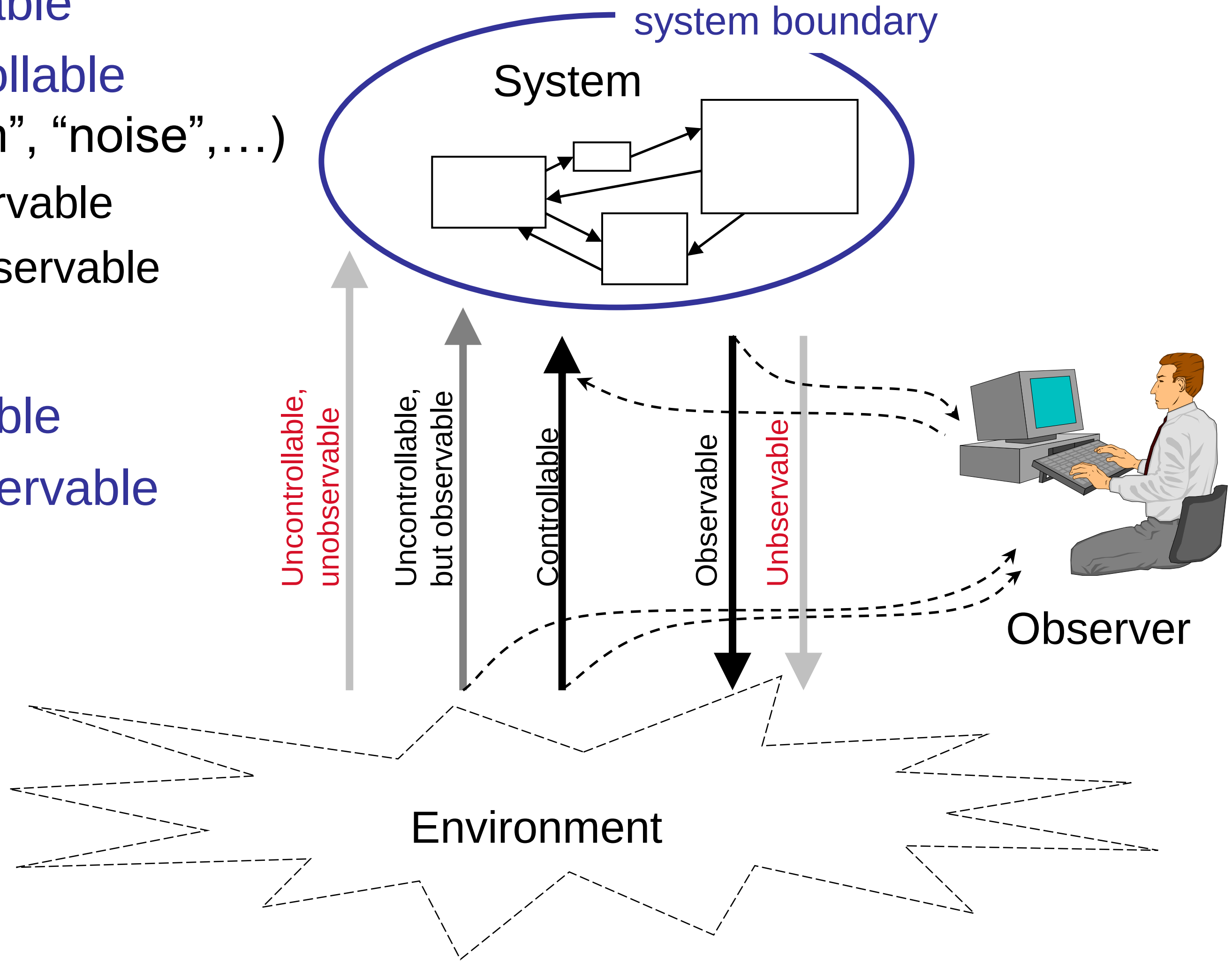
- Actually, a very vague notion—pretty much anything can be a system!
  - ‘A system is what is distinguished as a system.’ (Brian Gaines)
- A system is something that we want to see separated from its environment through an (arbitrarily chosen) **boundary:**
  - **Inside** the system:
    - [opaque, i.e., black box — less interesting]
    - or some structure, mechanisms, rules
    - or even sub-systems
  - (usual case)
  - **Outside** world (not part of the system!):
    - Environment, context
    - Interaction: Input from outside world, output into outside world



- A system has a **purpose**
  - Nobody defines something as a system without some purpose in mind

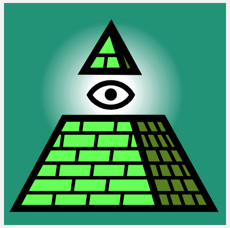
System, environment, observer

- Input
  - Controllable
  - Uncontrollable ("random", "noise",…)
    - Observable
    - Unobservable
- Output
  - Observable
  - Non-observable



Conceptualization

Conceptualization: Abstract, simplified view of the reference reality, represented for the purpose of performance and reliability evaluation.



Modeler / Observer

Abstraction

Description

Informal Reference Reality (Real-World System)

mimics

Stochastic Process (Semantic Model)

Queuing Network (Formal Syntactic Model)

Conceptual World

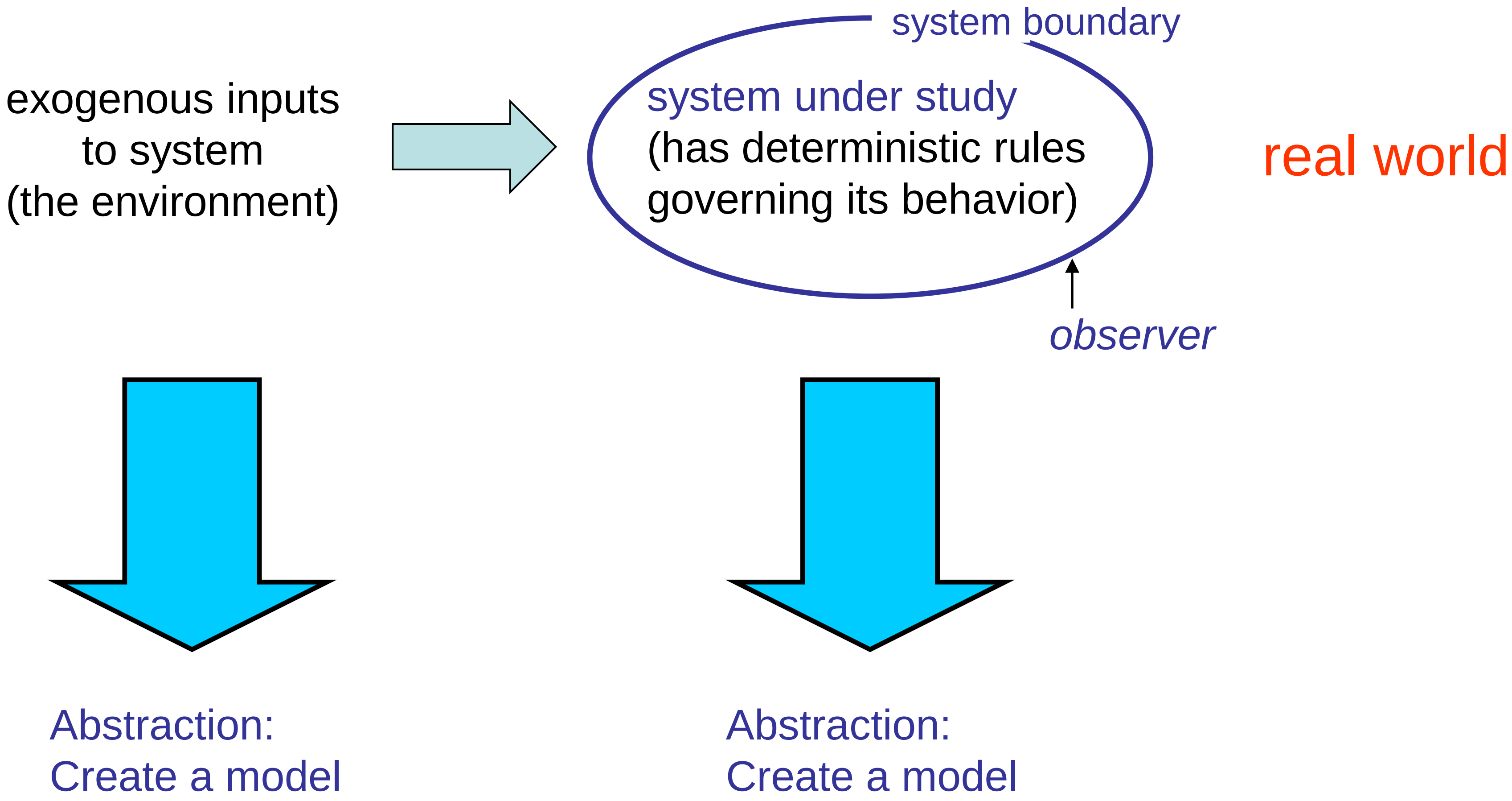
Conceptual and Real World

What is a model?

A **model …**

- is a system, too!
- mimics behaviour/characteristics of another system
- is material or immaterial
  - Material model: architecture models
  - Models we'll be talking about: normally immaterial
- allows experimental manipulation
- Purpose:
  - Simplification of original model: Reduction of complexity
  - Retaining those characteristics of original model that are important to the observer

Analysis of a system (actually: two!)



Ways to study a system

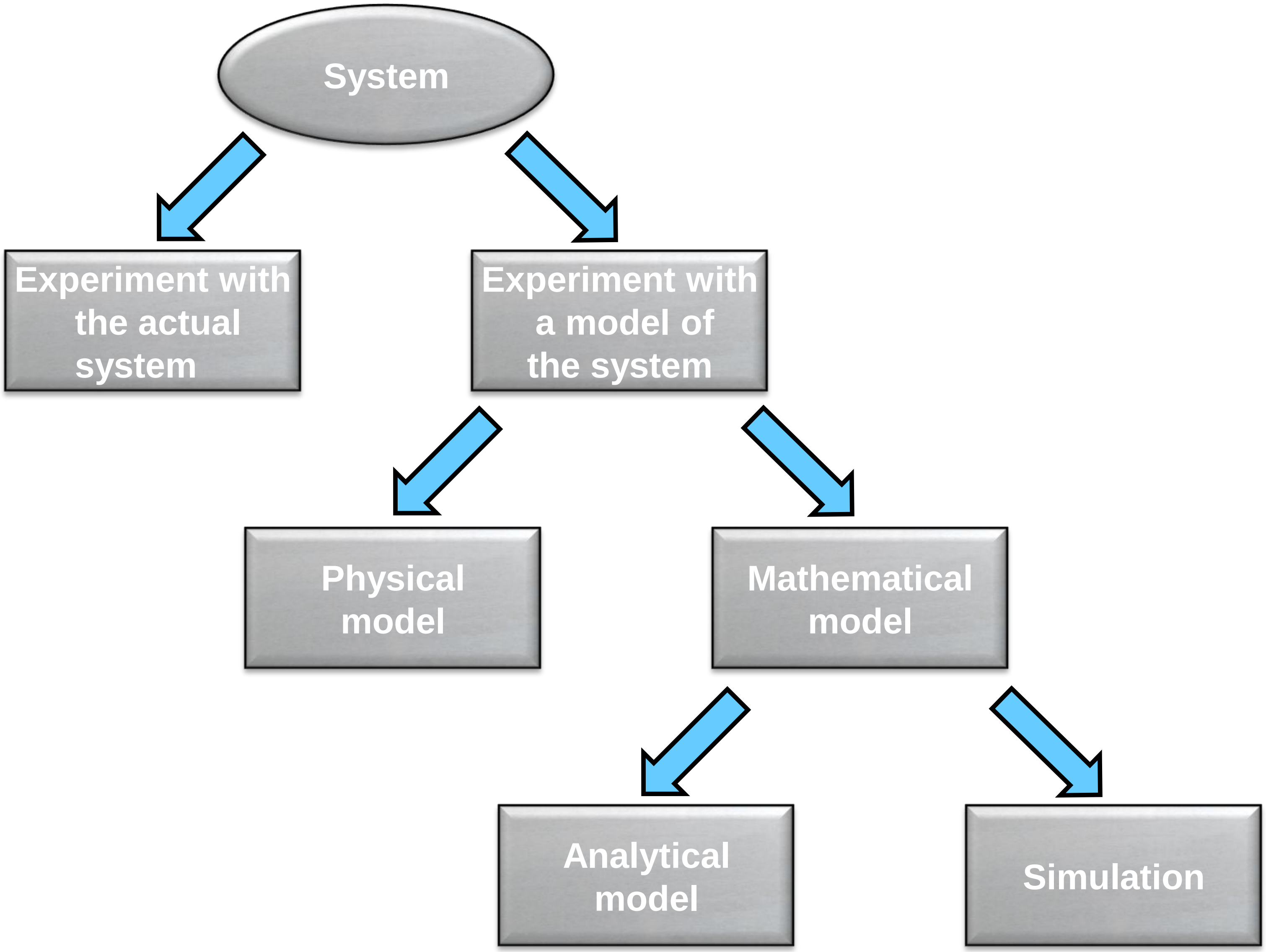


Figure adopted from LK 1.1

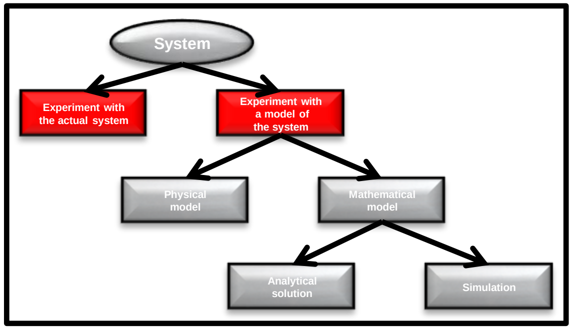
Ways to study a system (1/3)

Experiment with the actual system:

- Advantage:
  - Study is always valid
- Disadvantage:
  - Often too costly
  - Disruptive to the system
  - System might not even exist
  - Long-term study not feasible

Experiment with a model:

- Advantage:
  - Does not disrupt the actual system
  - No risks of system damage
- Disadvantage:
  - Accurate reflection of the actual system?
  - Is the model valid?



Ways to study a system (2/3)

Physical model:

- Advantage:
  - Often very accurate
- Disadvantage:
  - Usually expensive
  - Cannot be applied to all systems
  - Typically used for engineering or management systems
  - Smaller scales may result in different behavior

Mathematical model:

- Advantage:
  - Simple to apply
  - Allows abstraction of complex systems by using logical and quantitative relationships
  - Can be used for verification
- Disadvantage:
  - Accurate reflection of the actual system?
  - Is the model valid?
  - Are all relevant characteristics considered?

Ways to study a system (3/3)

Analytical solution:

- Advantage:
  - Often faster than simulation
  - Optimal for non-complex systems
  - Can be used for verification
- Disadvantage:
  - Complex systems are hard to describe by a mathematical model
  - Analytical solution usually have to apply higher levels of abstraction

Simulation:

- Advantage:
  - Simple to apply
  - Very flexible in terms of complexity
- Disadvantage:
  - Accurate reflection of the actual system?
  - Are all relevant characteristics considered?

Alternatives to simulation (1/2)

Evaluation spectrum:

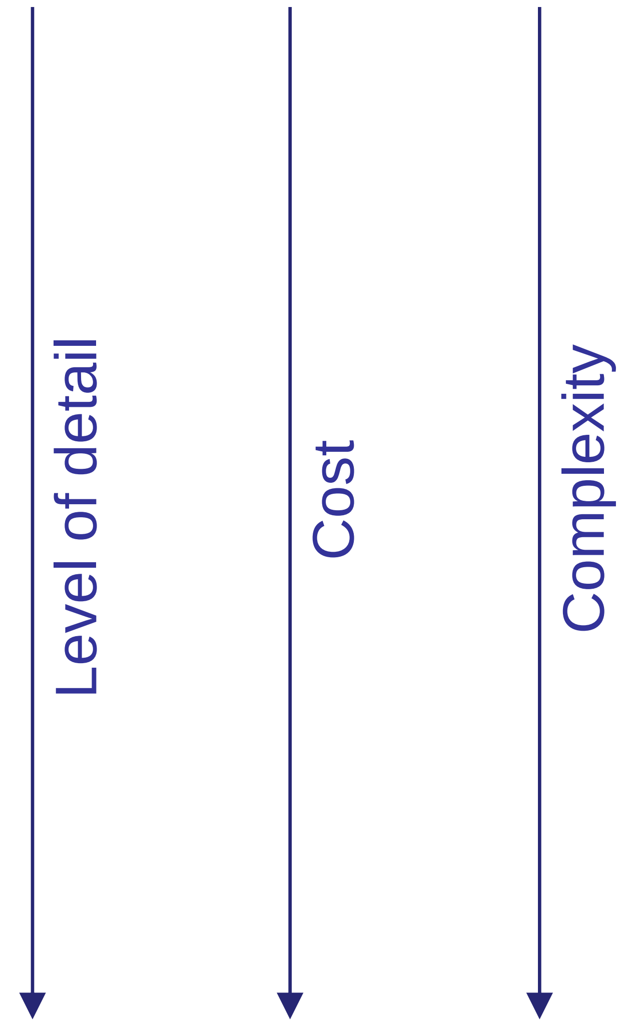
- **Purely mathematical model using closed-form expressions**
- Numerical models

less detailed

- Simulation

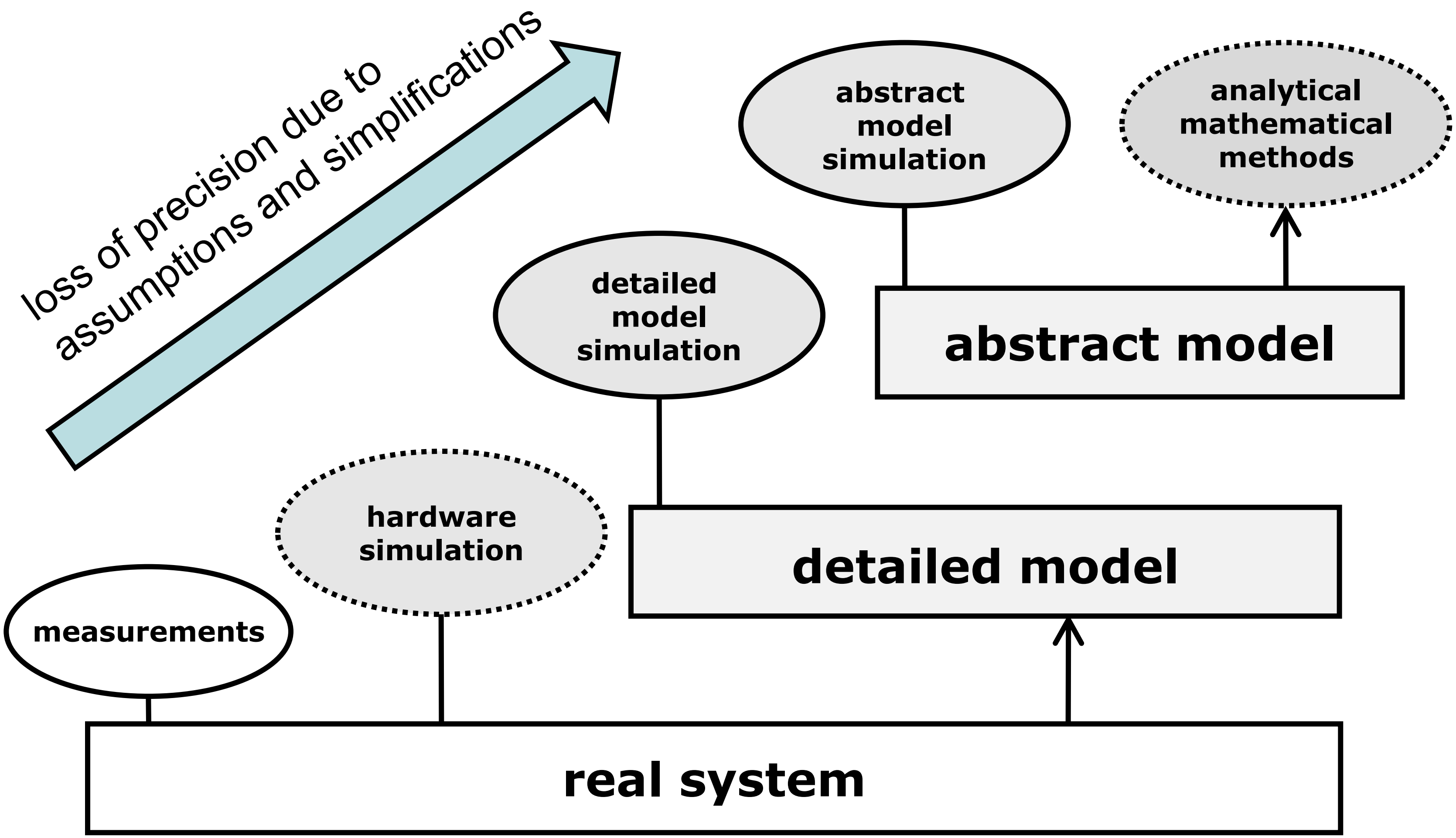
more detailed

- Emulation
- Prototype
- Operational system



Alternatives to simulation (2/2)

Evaluation spectrum:



When to use simulations (1/2)

- It may be too difficult, hazardous, or expensive to observe a real, operational system.
  Examples:
  - Virus epidemic
  - New routing protocol in the Internet
- There is only one real system, but we want to quickly evaluate alternatives and what-if scenarios.
  Examples:
  - Different router configurations
  - Different types of network traffic (realistic, low rate, full rate,…)

When to use simulations (2/2)

- Parts of the system may be unavailable / not be observable.
  Examples:
  - Internals of a biological system
  - Internals of a switch chip
- The original system runs on a very slow timescale, and/or we want to make predictions.
  Examples:
  - Climate predictions (10s to 1000s of years)
  - Milky way eating Sagittarius dwarf (100 mio years and more)
- It may be too difficult or intractable to model a system in detail using only closed-form expressions ("formulae").
  Examples:
  - Physical processes in atmosphere (weather, climate,…)
  - n-bodies problem, n≥3
  - Complex network with many TCP hosts

Simulation: Drawbacks

- Caution: Does model reflect reality? Or is it too oversimplified?
- Large scale systems = Lots of resources to simulate, especially if accurate simulation is required
- Large scale systems = Lots of resources for simulator:
  - May be slow (computationally expensive: 1 min real time could be hours of simulated time!)
  - May eat huge amounts of RAM
  - May write out gigabytes of output (…which needs to be analyzed after!)
- It's an art: determining right level of model complexity
- Statistical uncertainty in results:
  - Was the simulation accurate/detailed enough?
  - Are the observed effects just artifacts/statistical outliers? Remember: Some input comes from a (pseudo-)random generator!

When to use analysis

- Hard boundaries are required rather than statistics
- Non-complex systems
- Complex systems which can be divided intro smaller sub systems
- Immediate feedback is required
- Simulations would be to complex
- Systems which can be described by a model
  - Some systems can be to complex or cannot be simplified without significant loss of accuracy

Advantages of Modeling

- Save lives
- Save money
- Save time (?)
  - Buying hardware, connecting and configuring a huge test network takes longer than setting up a simulation (…usually)
- Development / Find bugs (in design) in advance
  - The earlier a bug is detected, the less its removal will cost
- More generally applicable than analytic/numerical techniques
- Detail: can simulate system details at arbitrary level

Use cases and applications for modeling

- Analyze systems before they are built
  - Reduce number of design mistakes
  - Optimize design
- Analyze operational systems
  - What-if scenarios
  - Find reasons for aberrant behaviour
- Create virtual environments for training, entertainment
  - Flight simulators, battlefield simulators
  - …in fact, almost all computer games are simulations!

Applications: System Analysis (focus of lecture!)

"Classical" application of simulation; here, focus is on "discrete event" simulation

- **Telecommunication networks** (focus of lecture!)
- Transportation systems
- Electronic systems (e.g., microelectronics, computer systems)
- Battlefield simulations (blue army vs. red army)
- Ecological systems
- Manufacturing systems
- Logistics

Focus is typically on planning, system design

Simulations may take a long time to run

Analysis of System Performance

IN2072

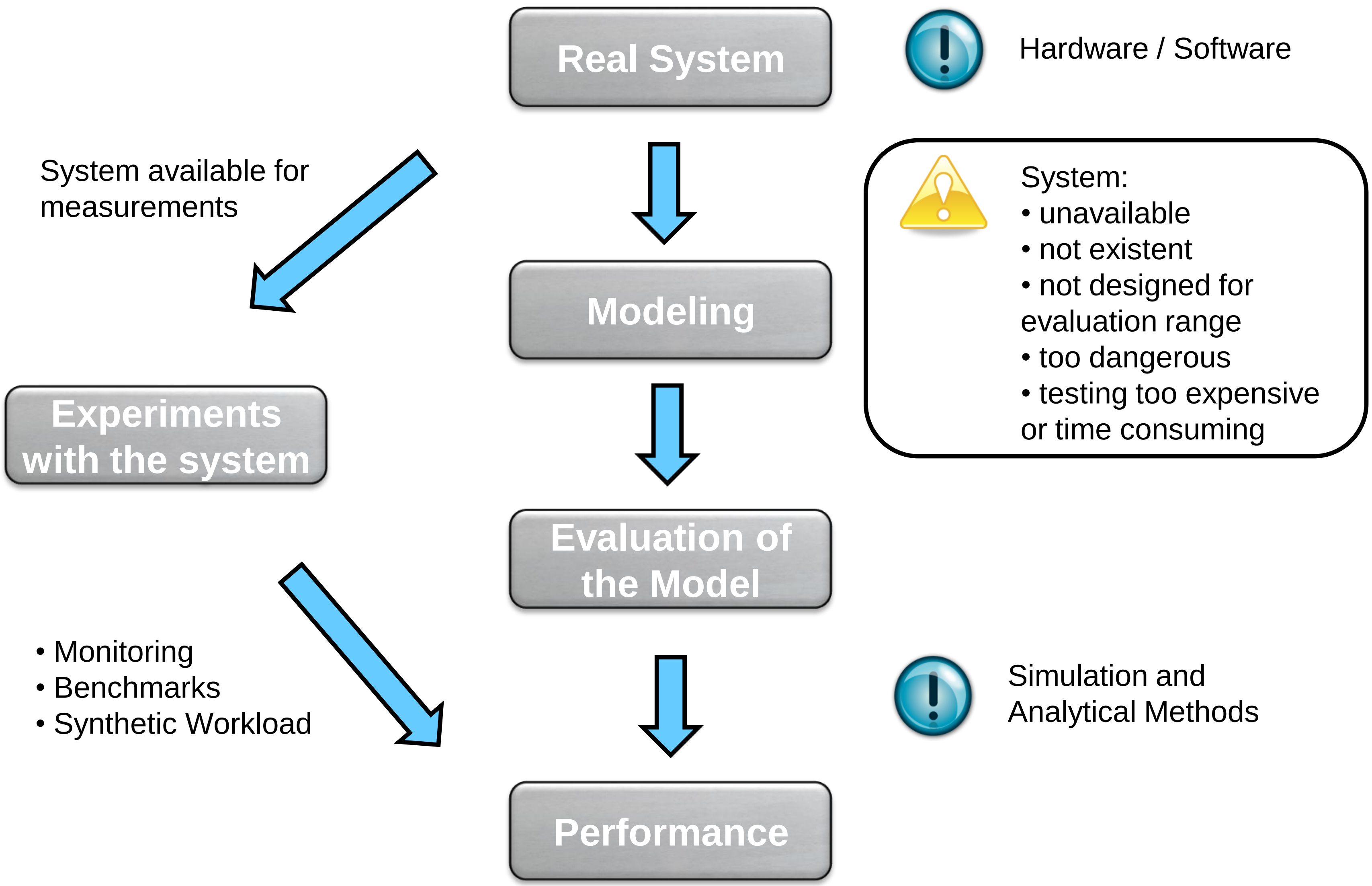
Chapter 0.a

Performance Modeling

Dr. Alexander Klein
Prof. Dr.-Ing. Georg Carle

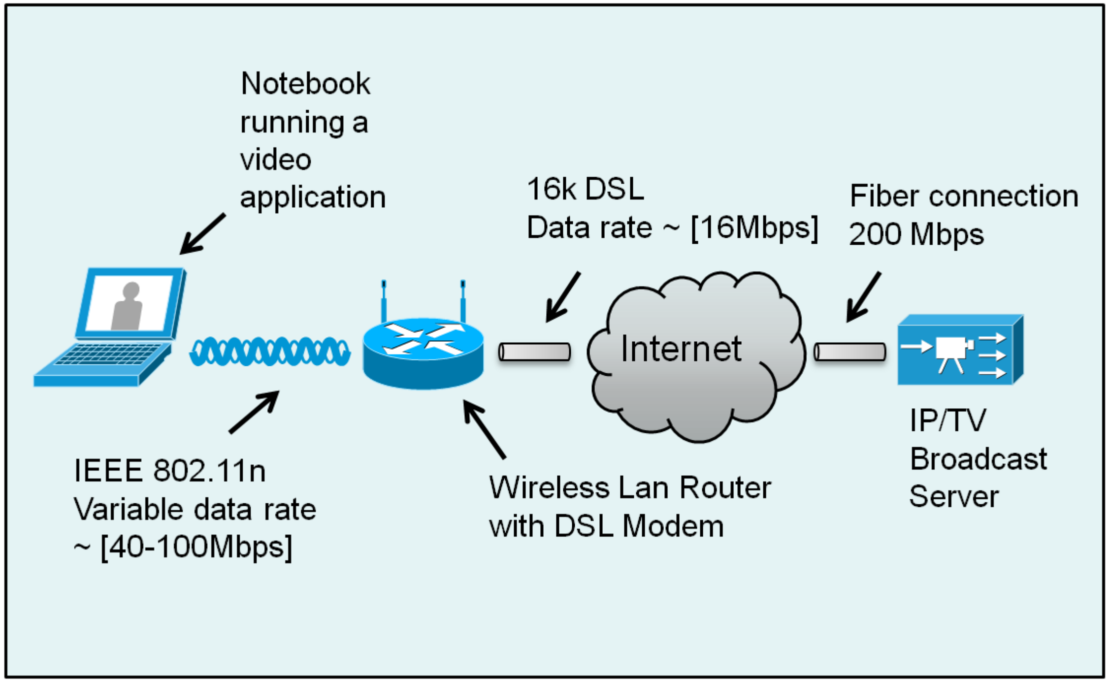
Chair for Network Architectures and Services
Department of Computer Science
Technische Universität München
http://www.net.in.tum.de

Steps of Performance Modeling



Performance Metrics

- Waiting time
- Service time
- Retention time
- Queue length
- Waiting probability
- Blocking probability
- Response time
- Utilization
- Throughput
- Speed up
- Deadlines

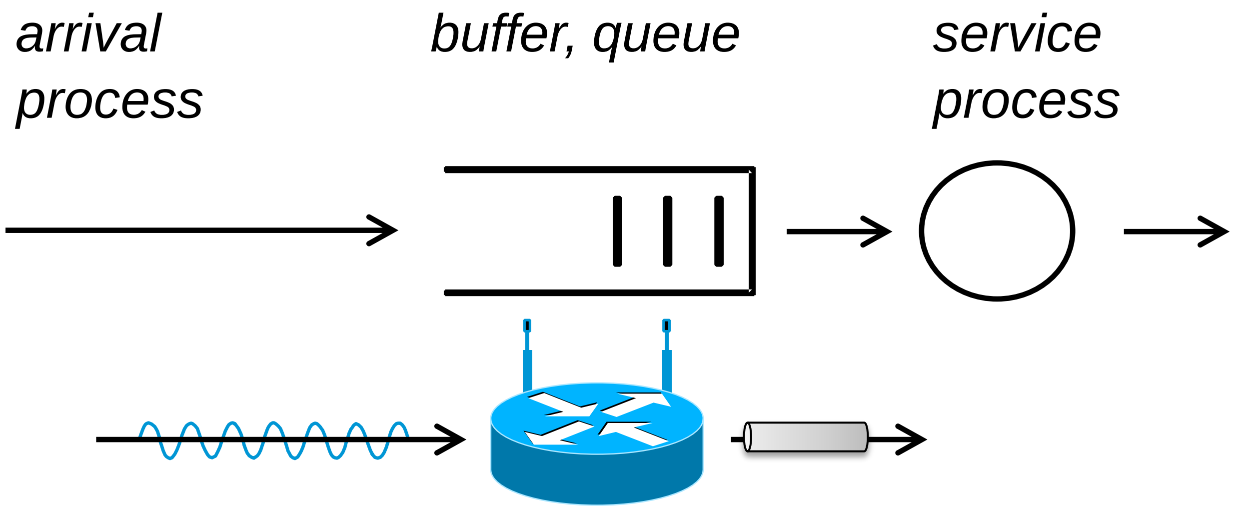


➡ Performance evaluation for given system configuration?

➡ Performance optimization by evaluating a set of variable system parameters?

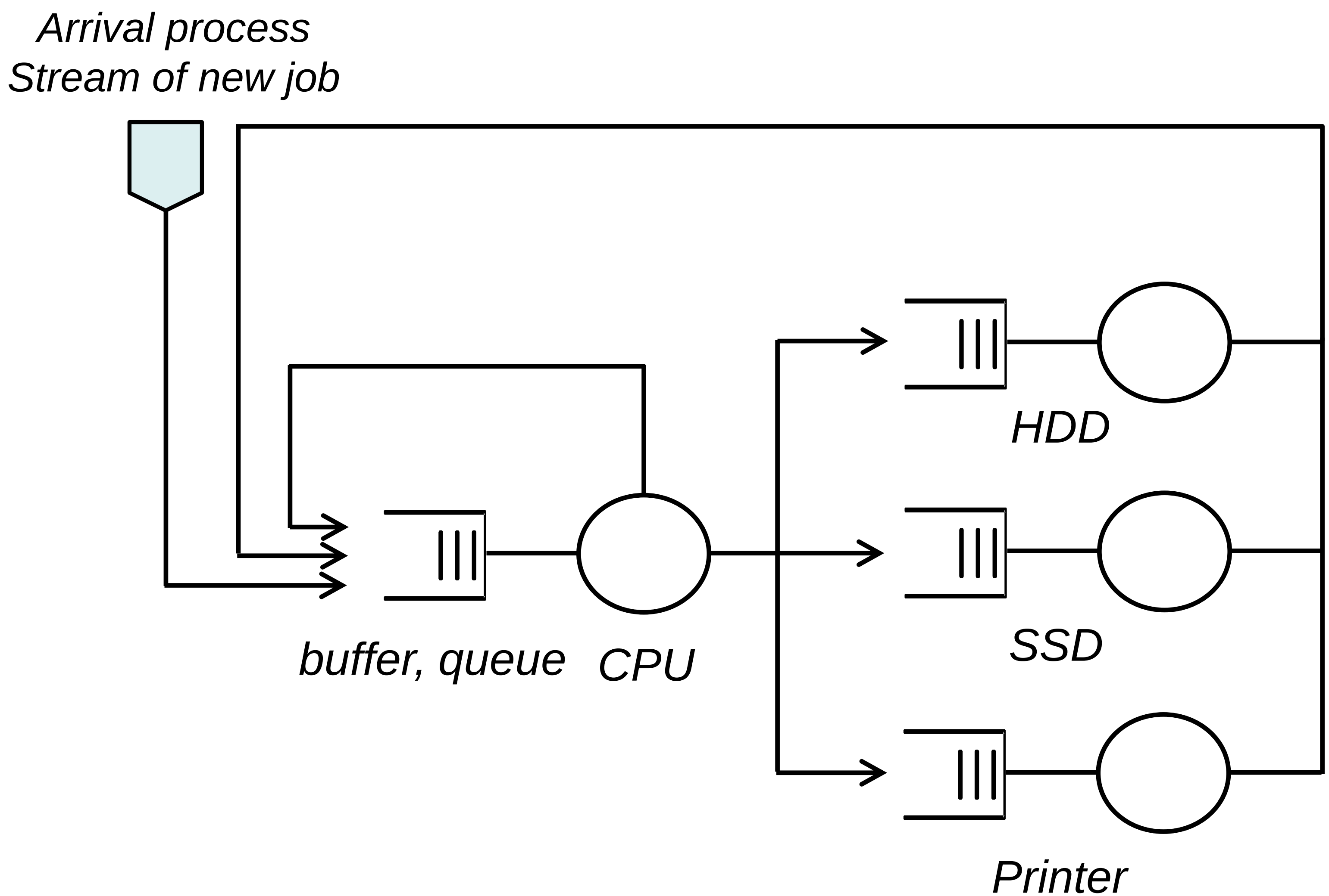
Model Types

- Queueing Network Model:



Model Types

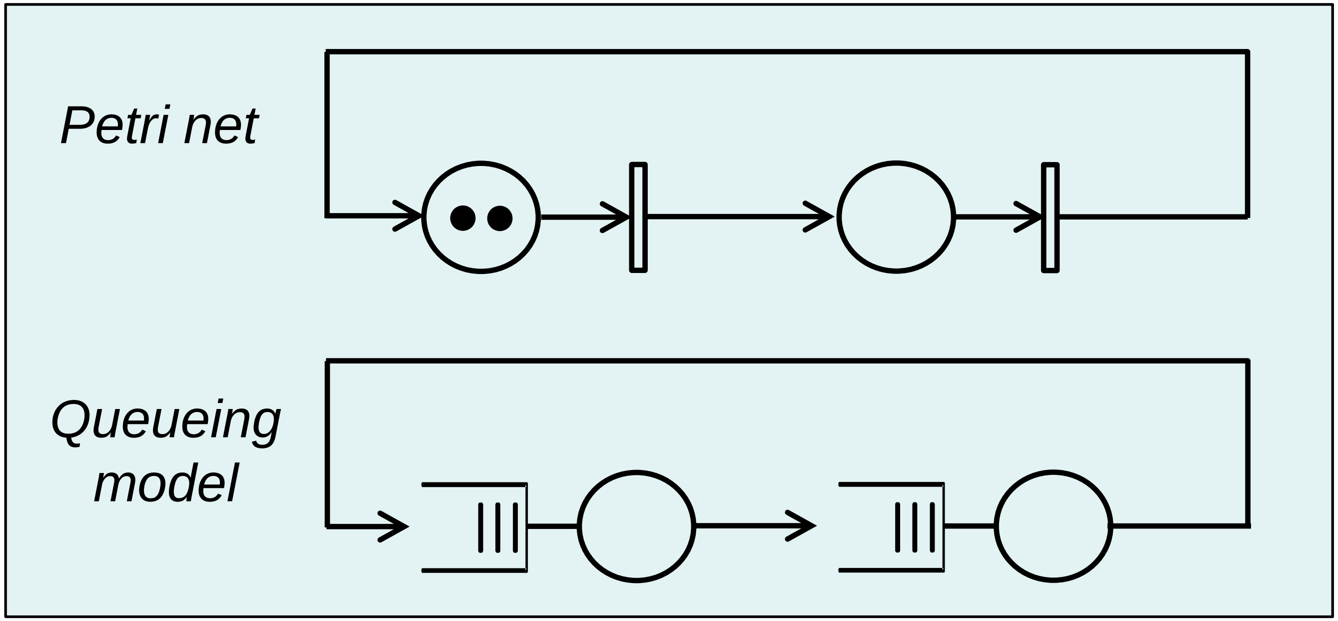
- Queueing Network Model:



Model Types

- Petri Net:
  - System described by:
    - Transitions
    - Places
    - Edges between transitions and/or places
    - Tokens in places

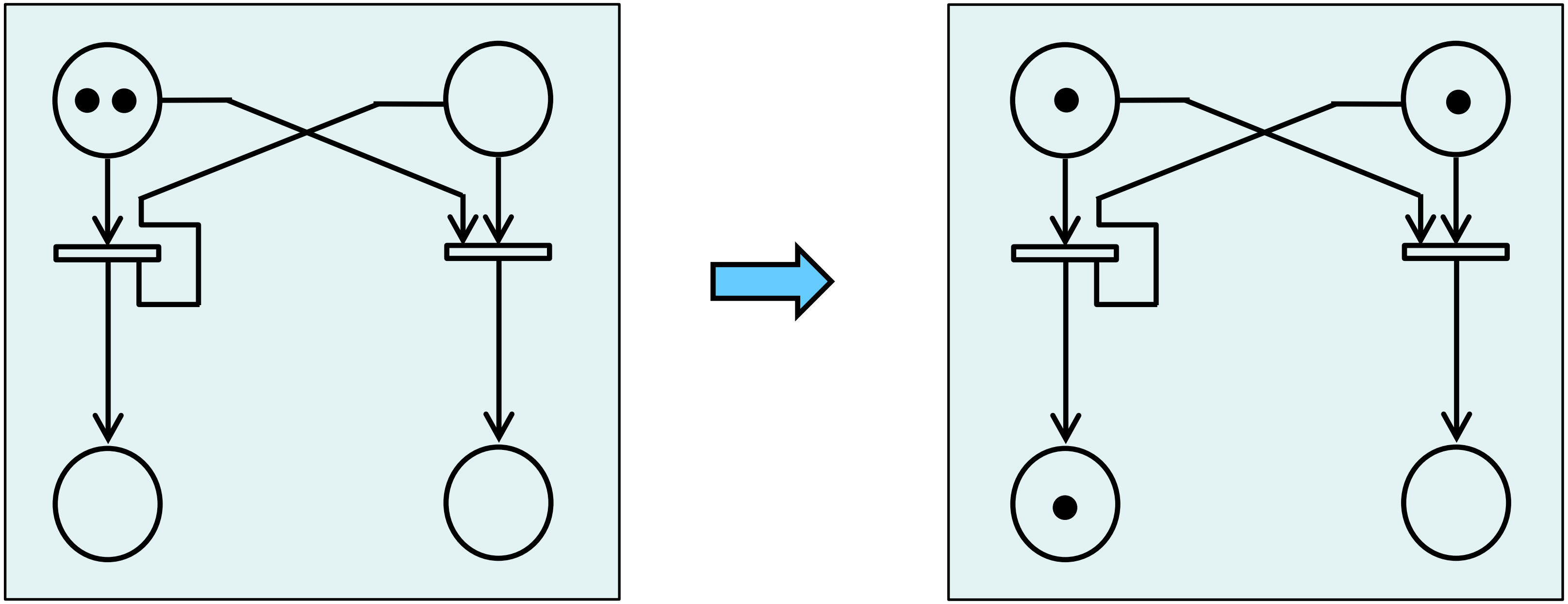
- Example (1/2):



Model Types

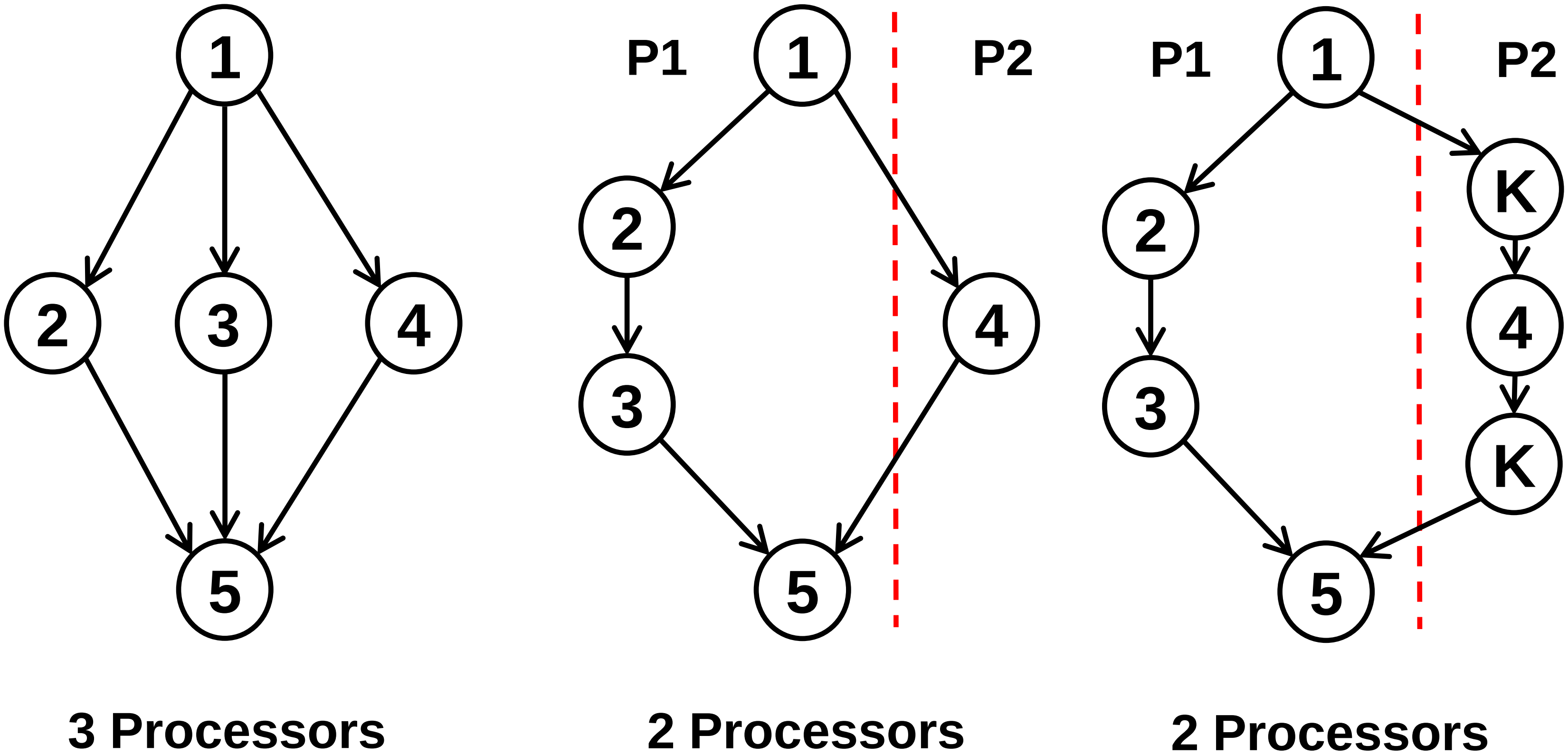
- Petri Net:

- Example (2/2):



Model Types

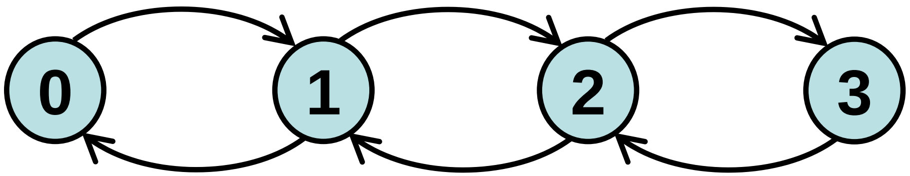
- Precedence Graph:
- Example: Task composed of 5 subtasks



Model Types

- Markov Model:
  - System described by:
    - States
    - Transitions between States

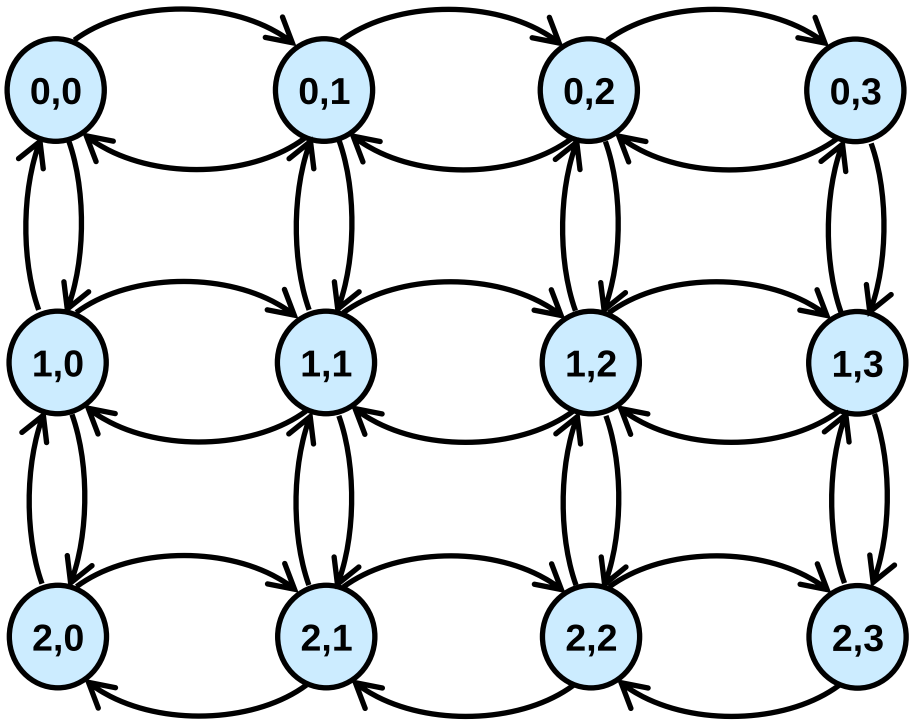
- Example (1/2):



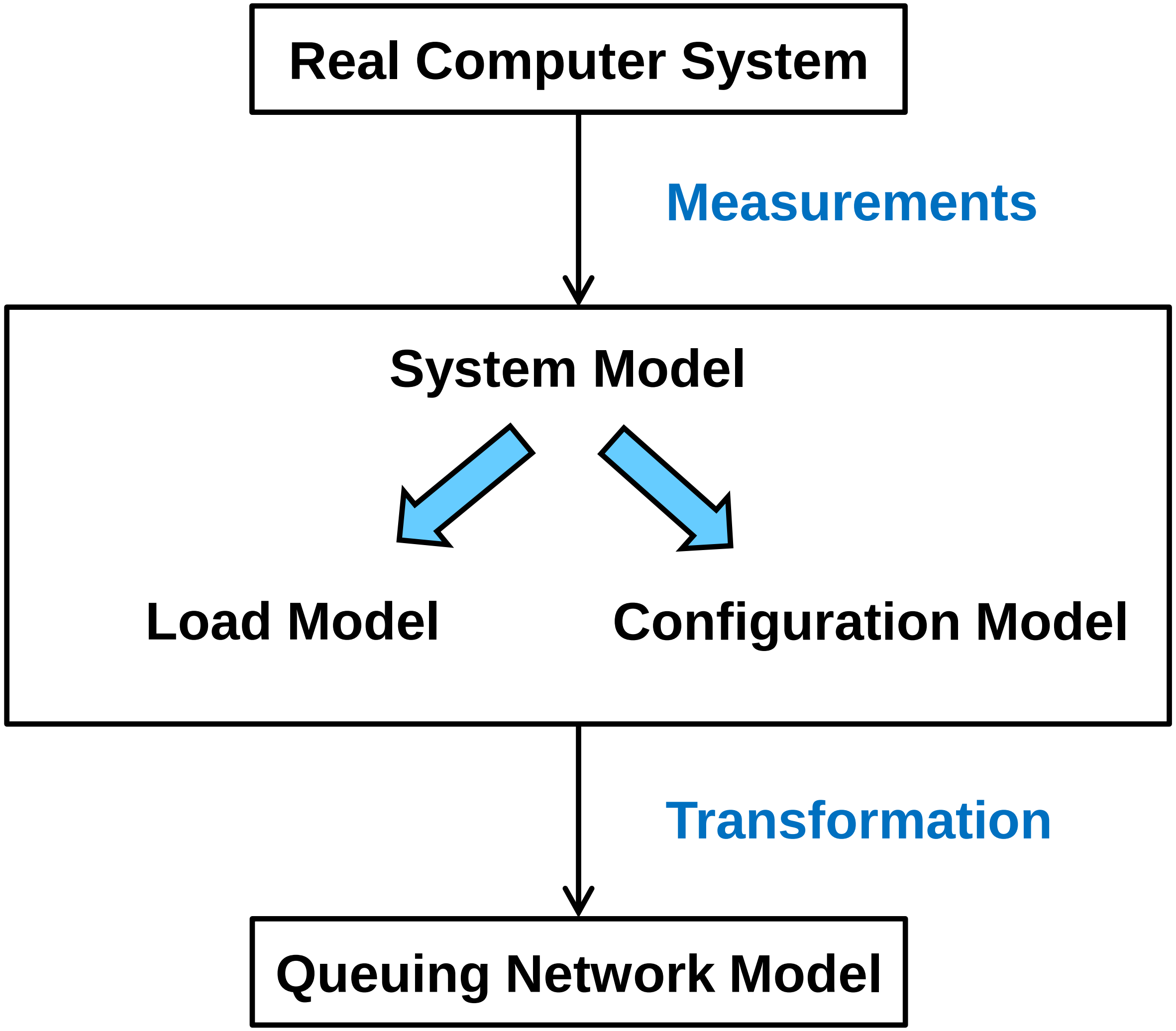
Model Types

- Markov Model:

- Example (2/2):



Modeling Process



Investigation of the Model

- Analytical Methods
  - Determination of the performance function F:
    - Performance F = F(Load, Software, Hardware)
      = F(System parameters)
    - Analytical Model
  - Advantage:
    - Influence of the system parameters is transparent
    - Optimization can be easily done
    - Short computing time
  - Disadvantage:
    - Complex systems can hardly be described by analytical models
    - Only an approximation of the real system

Investigation of the Model

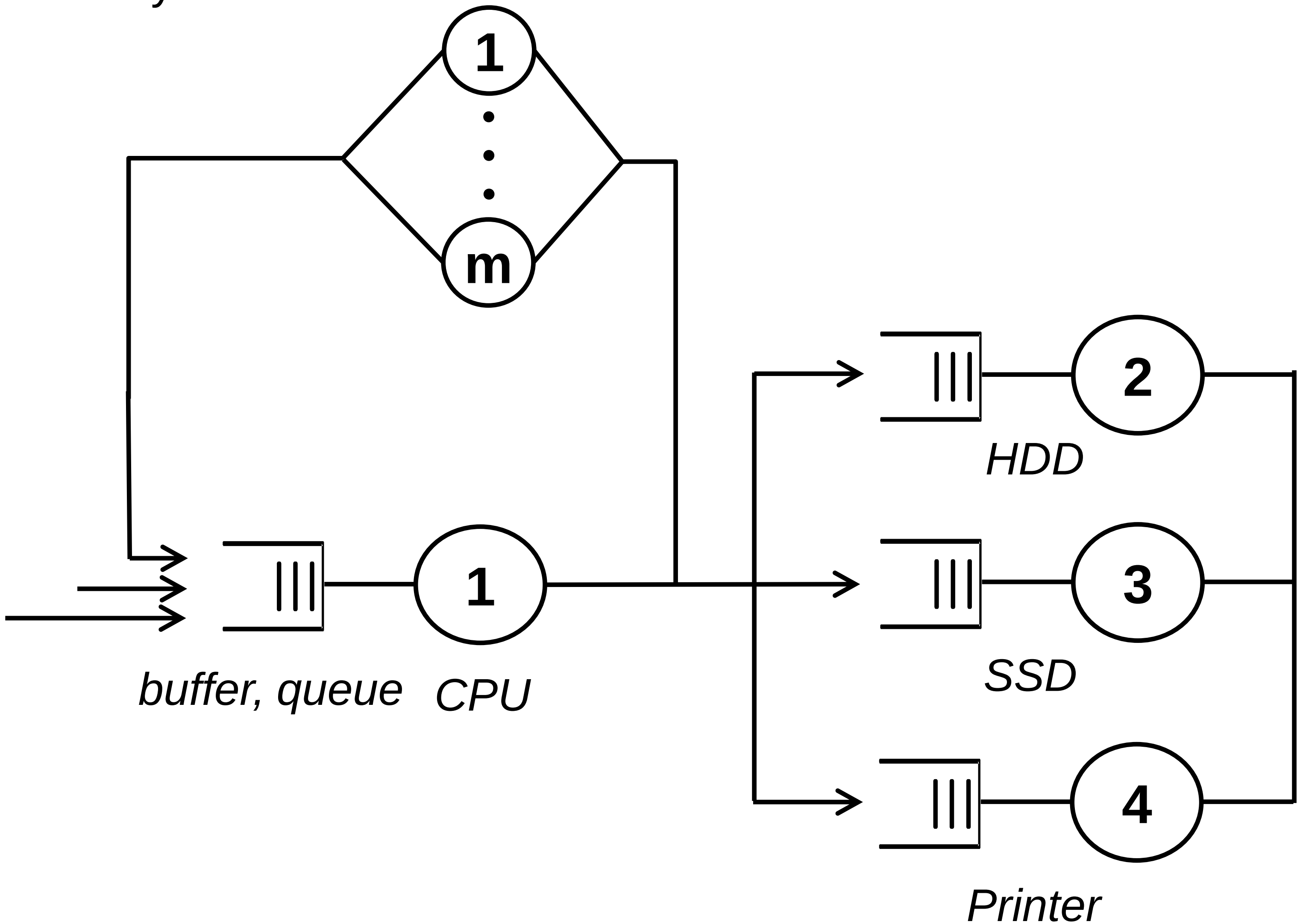
- Properties of the model:
  - Modeling and manipulation of the model is easier, cheaper and faster than the experiment with the real system
  - Model should (only) contain all relevant properties of the real system

Investigation of the Model

- Steps of Performance Evaluation
  - Computer system will be modeled using a queuing network model (or another appropriate model type)
  - Determination of the relevant system parameters (measurements, data from datasheets or estimation)
    - Interarrival times
    - Service times
    - Transition probabilities between components of the model
      - Distribution for state transitions
      - Mean value or variance is usually sufficient
  - Calculation of performance measures from the system parameters using formulas and algorithms, which will be presented and derived in this lecture.

Examples

- Terminal System



Analysis of System Performance

IN2072

Chapter 0.c

Queuing Network Model

Dr. Alexander Klein
Prof. Dr.-Ing. Georg Carle

Chair for Network Architectures and Services
Department of Computer Science
Technische Universität München
http://www.net.in.tum.de

Queuing Network Model

- What can be modelled?
  - Independet servers (CPU, devices, …)
  - Sequential allocation of servers by a job
  - Parallel allocation of different servers by different jobs

- What cannot be modelled?
  - Parallel allocation of multiple (different) resources (CPU + memory)
  - (Dynamic) Memory constraints
  - Blocking
  - Synchronisation

Queuing Network Model

What are we talking about… and why?

- Simple queue model:
  - Customers arrive at random times
  - Execution unit serves customers (random duration)
  - Only one customer at a time; others need to queue

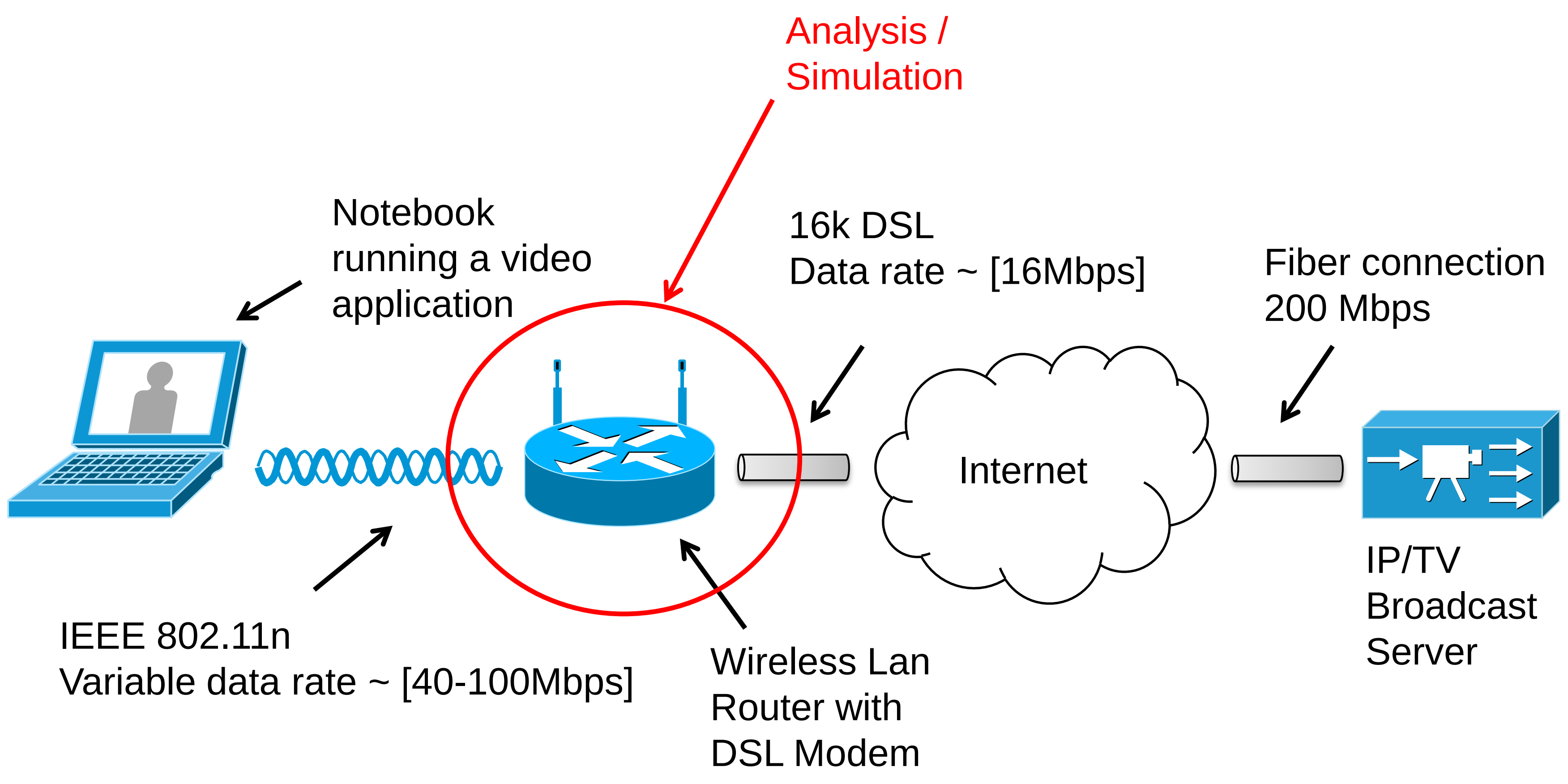
- Standard example

- Give deeper understanding of important aspects, e.g.
  - Random distributions (input)
  - Measurements, time series (output)
  - …

Queuing Network Model

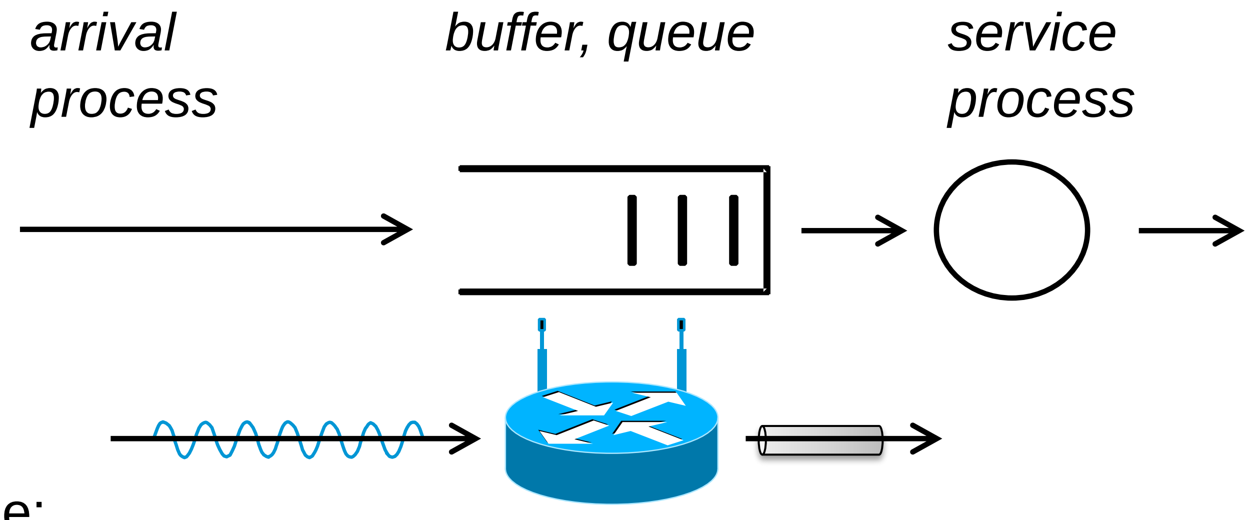
- Example:

  Computer networks ⟶ Router



Queuing Network Model

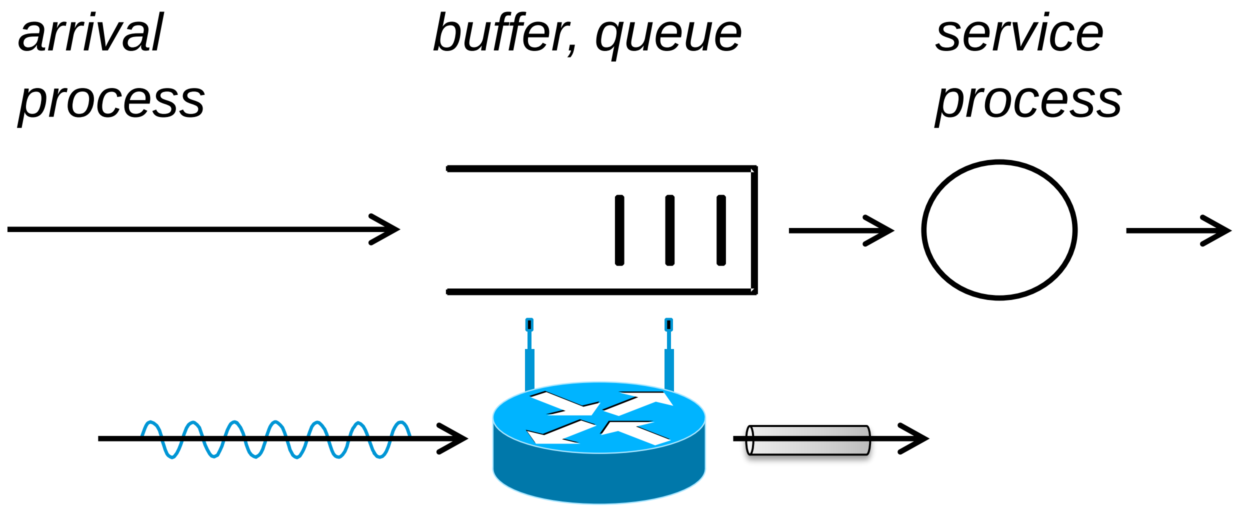
Waiting Queue Theory



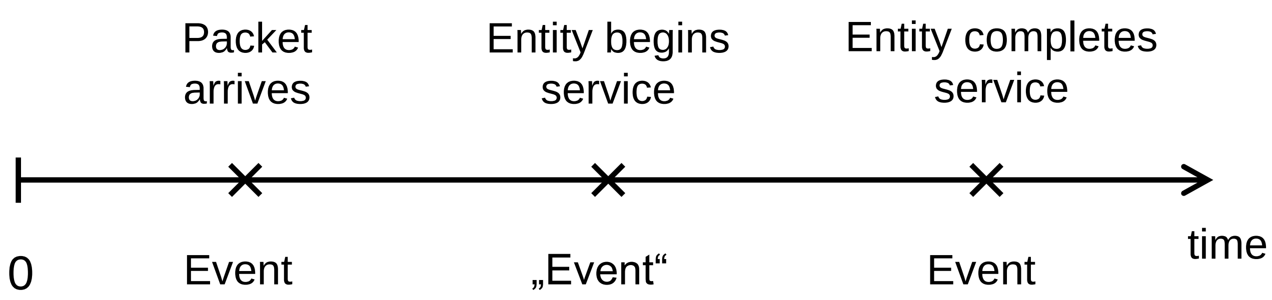
- Example:

  Router
  - Data packets arrive at the router via its wireless interface
  - A packet is forwarded immediately via the DSL interface if the buffer is empty and no packet is currently transmitted
  - Otherwise the packet is stored in the buffer if the buffer is below its maximum capacity
  - The service process simulates the time that is required by the router to write a packet on the DSL interface

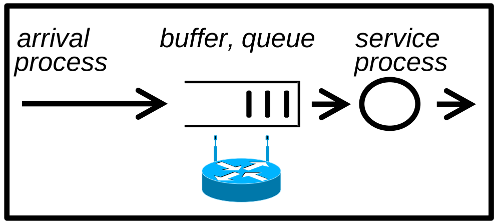
Waiting Queue Theory



- Entity/Packet flow:



Queuing Network Model



Router / System

- State variables:
  - Fill state of the queue (discrete) between [0 ; S] with S being the maximum queue capacity
  - State of the service process (discrete)
    - Idle (0)
    - Busy (1)

- Events:
  - Packet arrival:

    A new packet arrives at the router

    Process:

    - Increase queue by one if service process is busy and queue size is below maximum capacity
  - Service completion:

    Service process has transmitted a packet

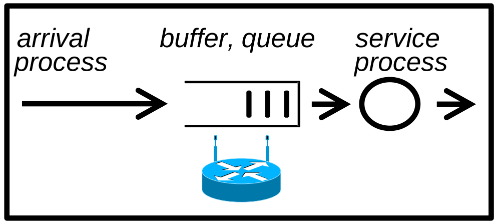
Queuing Network Model

- Events:
  - Packet arrival:

    A new packet arrives at the router

    Process:

    - Increase queue by one if service process is busy and queue size is below maximum capacity .
    - If queue size is at its maximum capacity drop the packet.
    - If service process is idle, set service process to busy state and schedule the next service completion event.

  Router / System

  - Service completion:

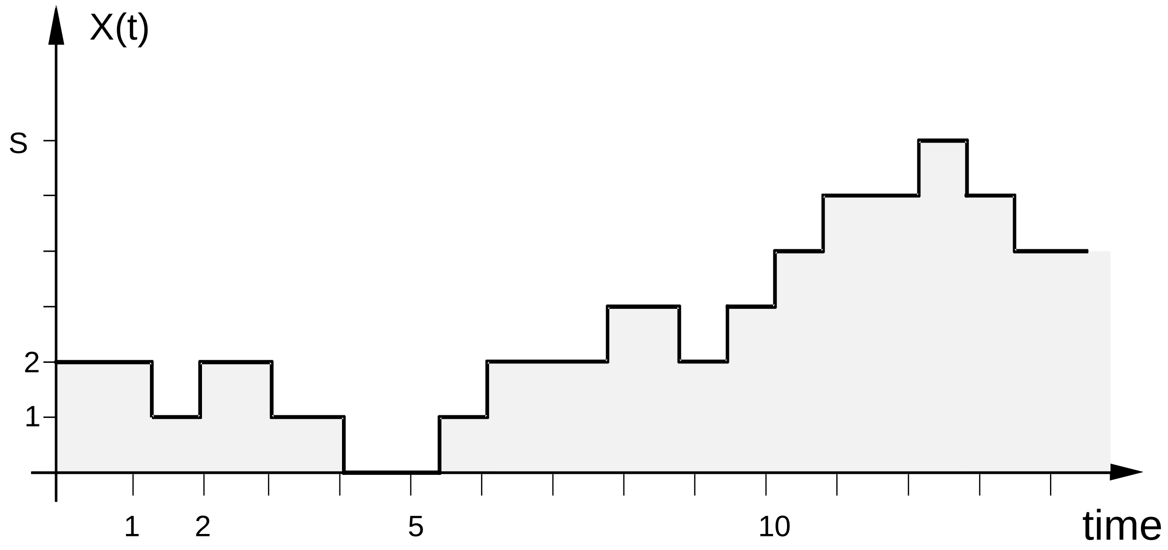
    Service process has transmitted a packet

    Process:

    - If queue size is equal to zero, set service process to idle.
    - If queue size is greater than zero, reduce the queue size by one and schedule the next service completion event.

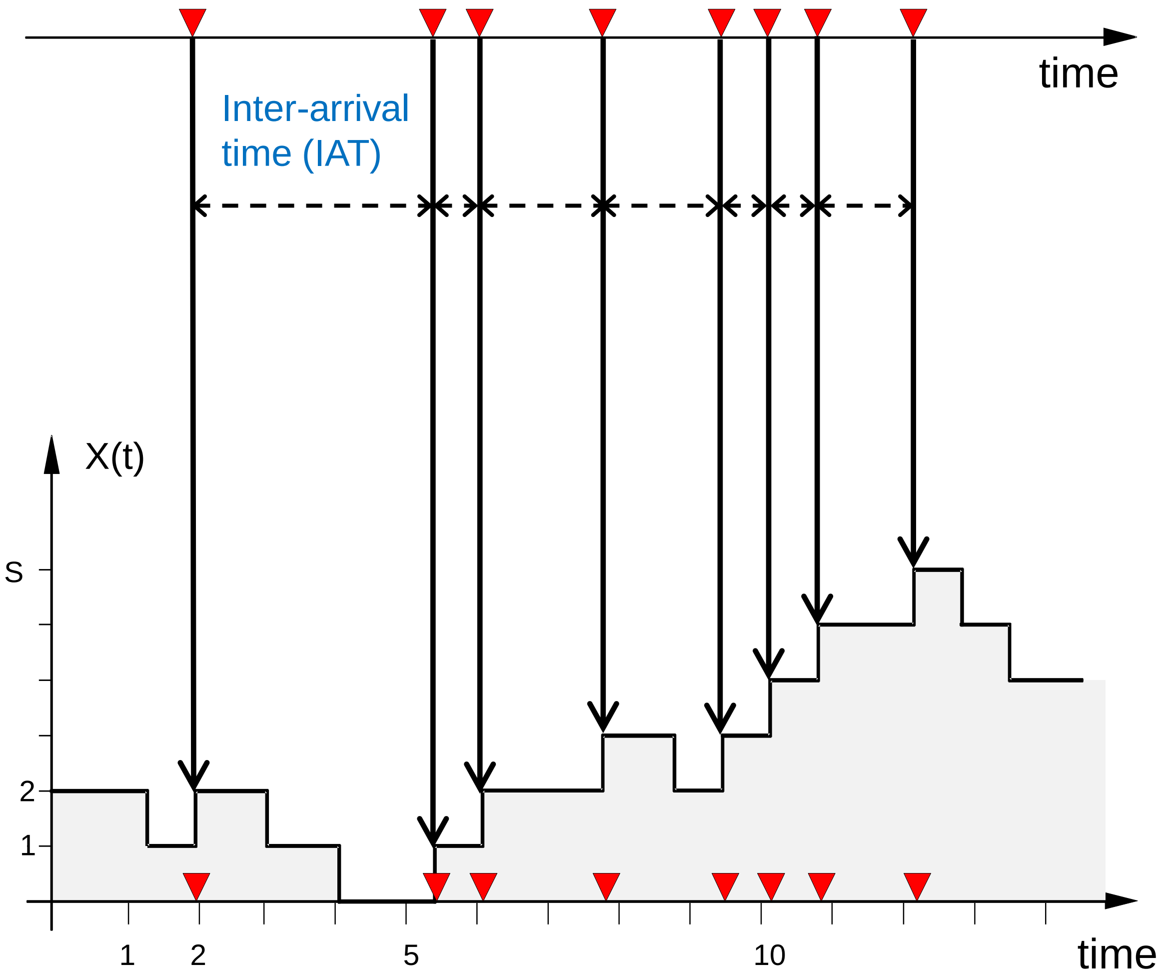
Queuing Network Model

- System characteristic:
  - Fill state of the queue at time is given by X(t)
  - The fill state of the queue can only change when an event occurs



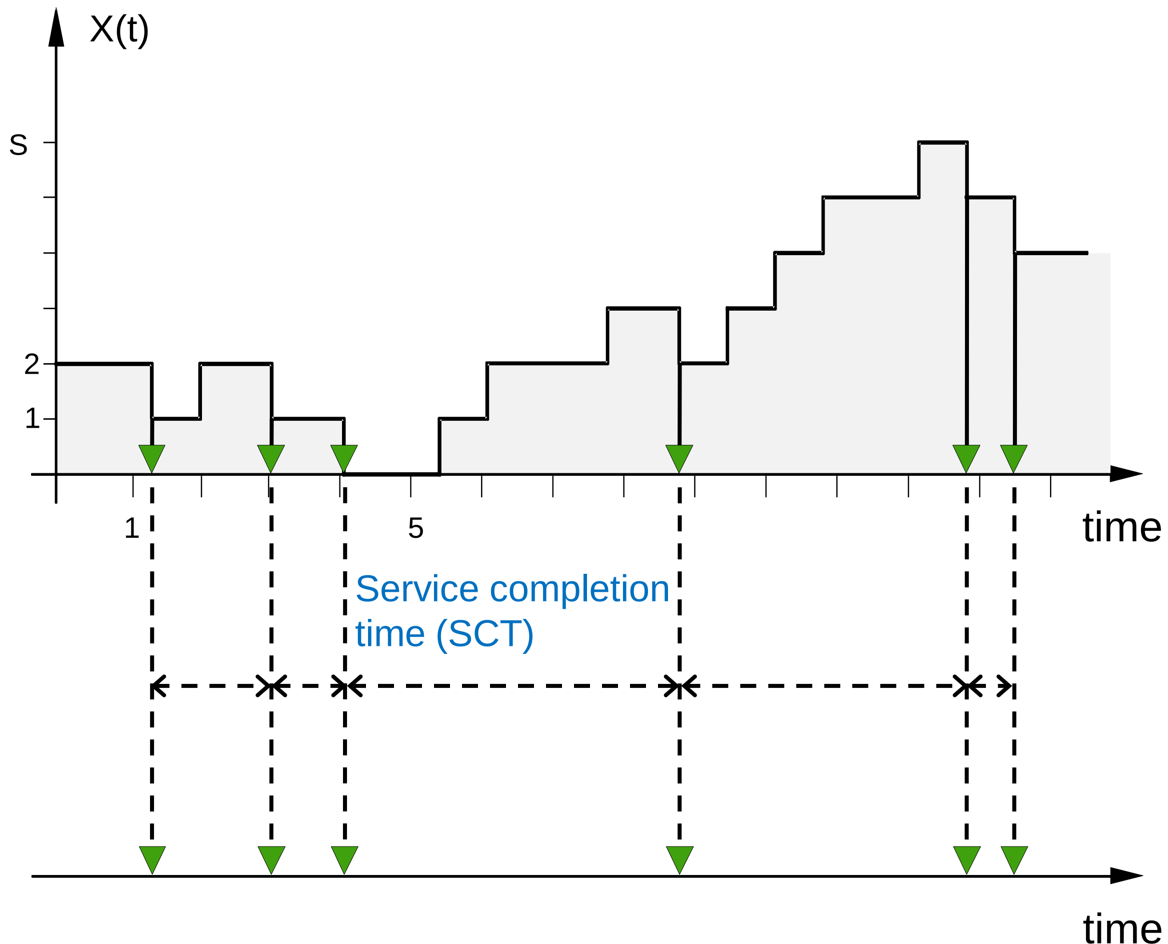
Queuing Network Model

- Arrival Events:



- Inter-arrival time: Time between consecutive arrival events

- Service completion events:

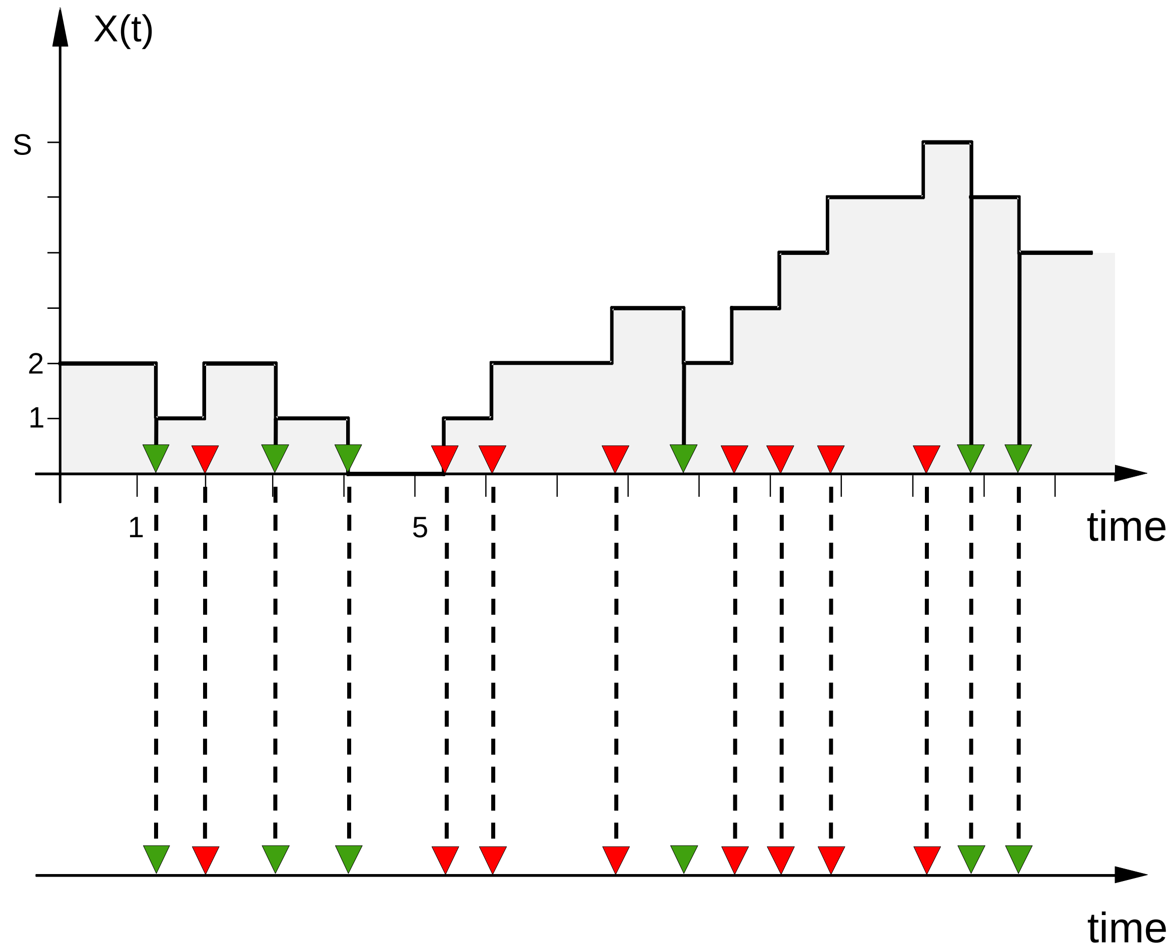


- Service completion time:

Time between consecutive service completion events.

Queuing Network Model

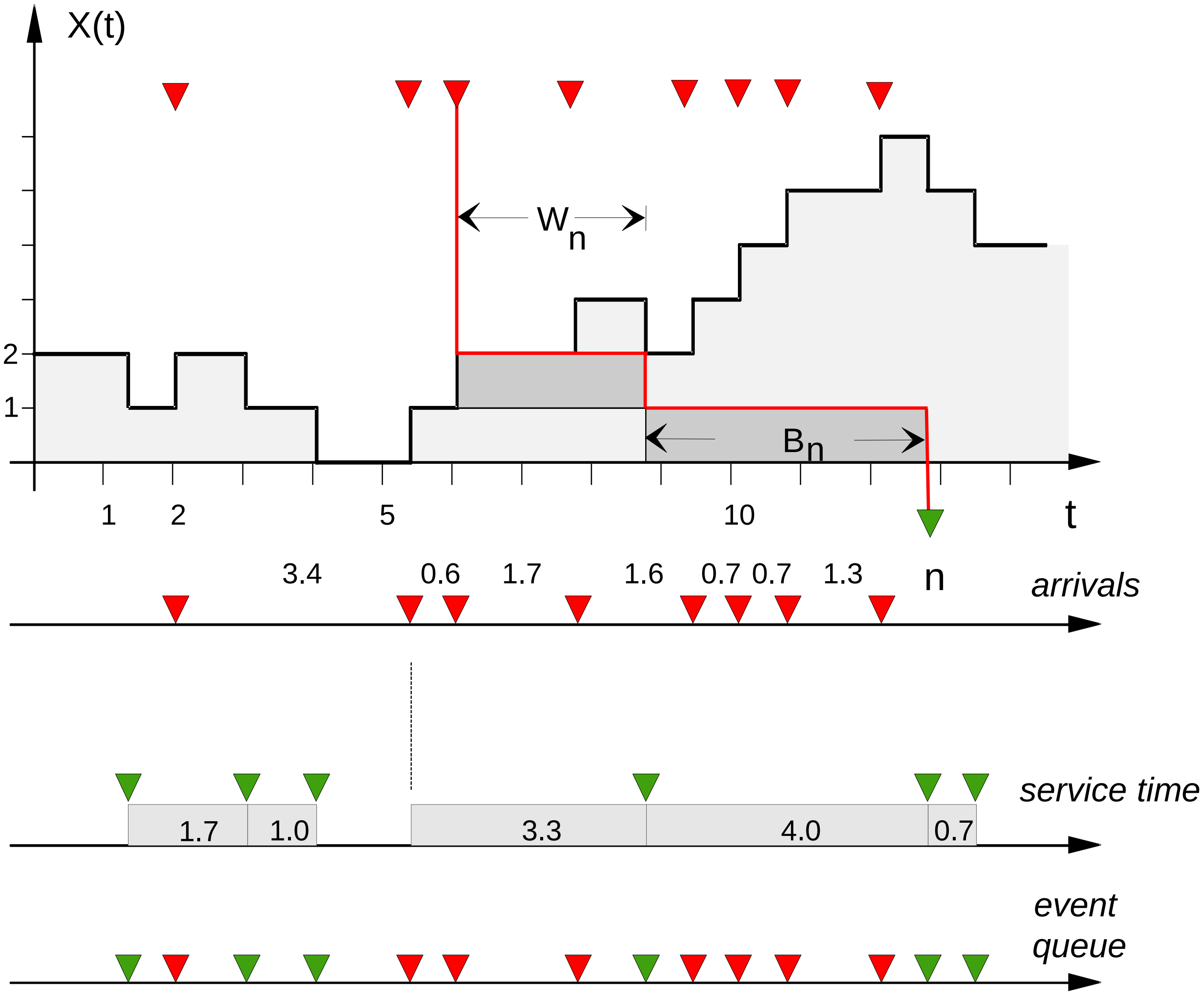
- Event queue:



Event queue is a dynamic list of events which is executed in sequential order.

Queuing Network Model

- Event queue:



What's inside a DES? (1/2: data)

- Simulated time: internal (to simulation program) variable that keeps track of simulated time
  - May progress in huge jumps (e.g., 1ms, then 20s, then 2ms,…)
  - Not related to real time or CPU time in any way!
- System state: variables maintained by simulation program define system state, e.g.: number of packets in queue, current routing table of a router, TCP timeout timers, …
- Events: points in time when system may changes state
  - Each event has an associate event time
    - e.g., arrival of packet at a router, departure from the router
    - precisely at these points in time, the simulation must take action (i.e., change state and maybe come up with new future events)
  - Model for time between events (probabilistic) caused by external environment
- Event queue: dynamic list of events (→later slides)
- Statistical counters: used for observing the system

What's inside a DES? (2/2: program code)

- Timing routine:
  - determines the next event and
  - moves the simulation clock to the next event time
- Event routine: "process the event", i.e., change the system state when an event happens
  - One subroutine per event type
- [P]RNG library routines: generate random numbers
- Report generators: compute performance parameters from statistical counters and generate a report. Runs at simulation end, at interesting events, and/or or at specific pseudo-events
- Main program:

```
while(simulation_time < end_time)
{
        next_event = getNextEvent();
        next_event.process();
}
```

Analysis of System Performance

IN2072

Chapter 0.d

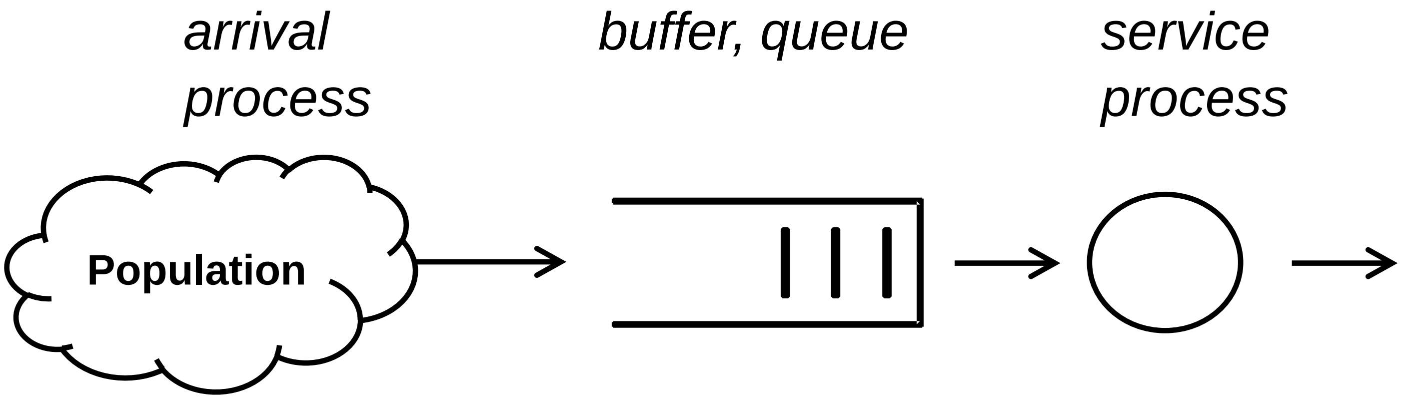
How to Simulate a Queuing Network

Dr. Alexander Klein
Prof. Dr.-Ing. Georg Carle

Chair for Network Architectures and Services
Department of Computer Science
Technische Universität München
http://www.net.in.tum.de

Discrete Event Simulation – Queuing Systems

- System Characteristics:



- Finite / Infinite
- Entity size
- Deadline

- Single queue / Multiple queues
- Finite / Infinite capacity
- Queuing strategy (FIFO, LIFO…)

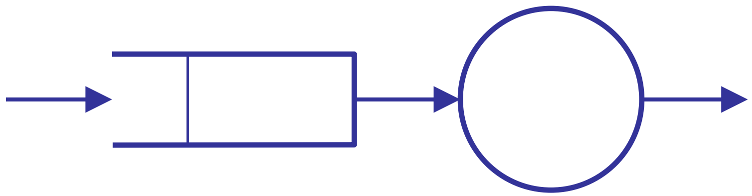
- Single processor / multiple processor
- State-(in)dependent
- Vacation

- Performance Characteristics:
  - Average/maximum customer waiting time
  - Average/maximum processing time of a customer
  - Average/maximum retention time of a customer
  - Average/maximum number of customers in the queue
  - Customer blocking probability
  - Utilization of the system / individual processing units

Waiting Queue Example

Queuing model: Input and output

- Input:
  - (Inter-)arrival times of customers (usually random)
  - Job durations (usually random)



- Direct output:
  - Departure times of customers
- Indirect output:
  - Inter-arrival times for departure times of customers
  - Queue length
  - Waiting time in the queue
  - Load of service unit (how often idle, how often working)

Waiting Queue Systems

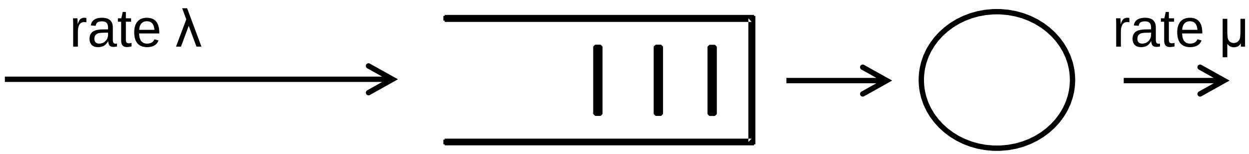
Applications:

System	Entity	Server
Store	Goods	(Lazy) cashier
Manufacturing	Customer order	Machine
Bank	Client	Clerk
Hospital	Patient	Doctor
Computer	Job	CPU
Computer network	Data packet	Radio channel
Cache	Content	Storage

Waiting Queue Theory

arrival process | *buffer, queue* | *service process*

rate λ | | rate μ

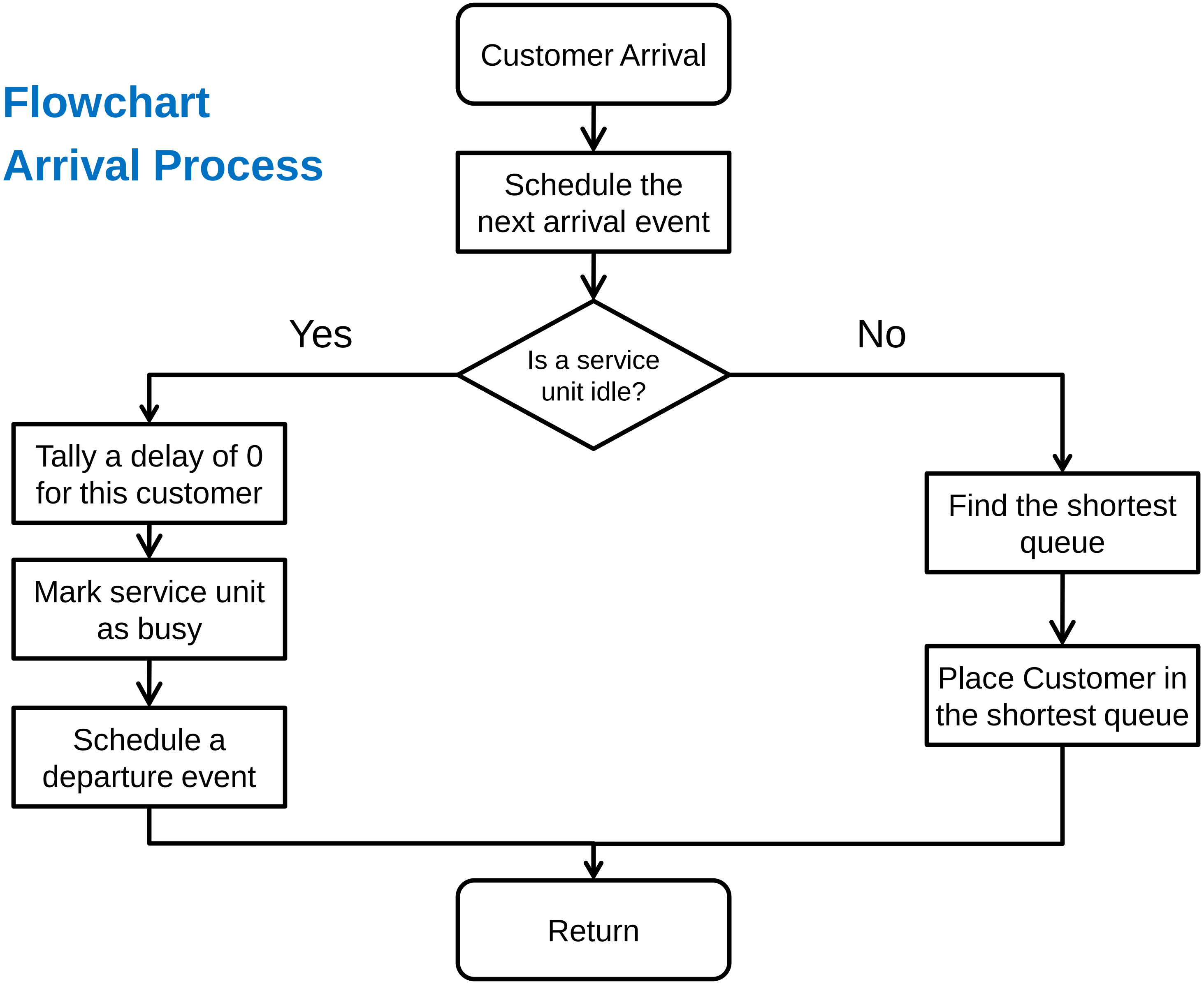


inter-arrival times		service times
3.4		1.7
0.6		1.0
1.7		3.3
1.6		4.0
0.7	random numbers	0.7
0.7		...
1.3		

Waiting Queue Example

Flowchart

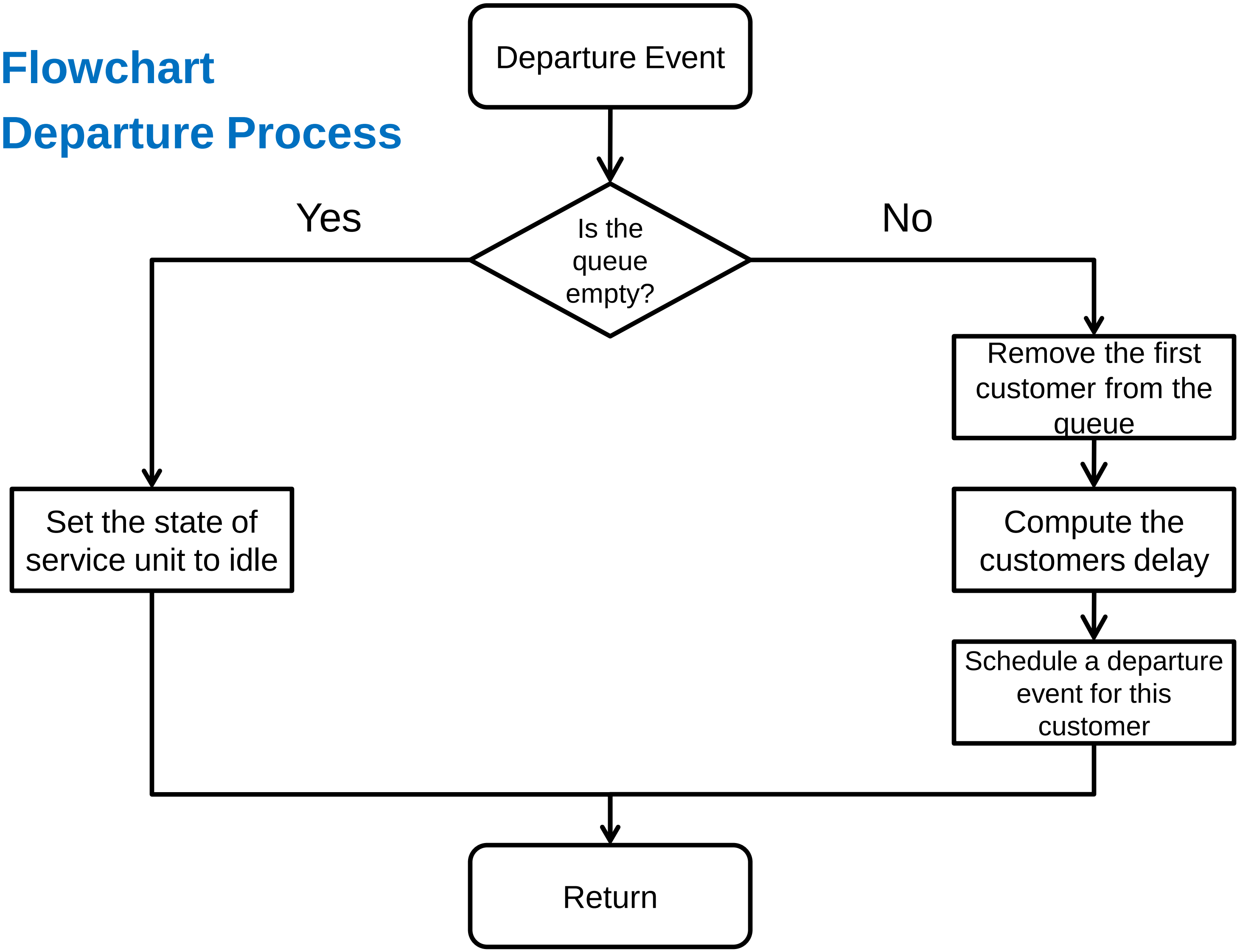
Arrival Process



Waiting Queue Example

Flowchart

Departure Process



Discussion

- How to build a Discrete Event Simulator?
- What are the necessary modules?
- What are the interesting performance parameters?

Step by Step

Waiting Queue Simulation – Step by Step

- Events:
  - Customer Arrival Events
  - Service Completion Events
  - Simulation Termination Event

- System variables:
  - Q(t) – number of waiting customers at time t
  - B(t) – number of busy servers at time t

- Performance parameters:
  - Average customer waiting time
  - Utilization of the system
  - Customer blocking probability

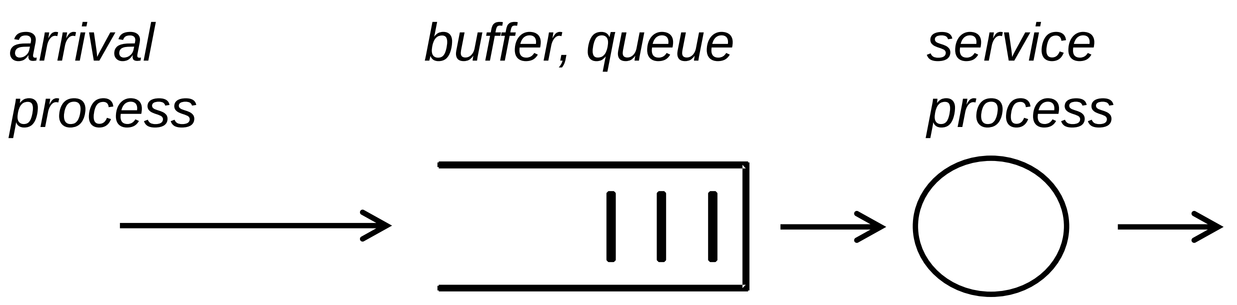
→ We will need to collect some data to provide these statistics

Waiting Queue Simulation – Step by Step

Waiting Queue Simulation:

- System with 2 serving units/processors
- Single queue with a capacity of 3 customers
- Simulation duration of 30 simulation ticks
- The first customer is inserted at time 0

System Characteristics:



- General independent inter-arrival times

- Finite capacity – 3 Customers
- Queuing strategy - FIFO

- Multi-processor – 2 processors
- General independent service completion times

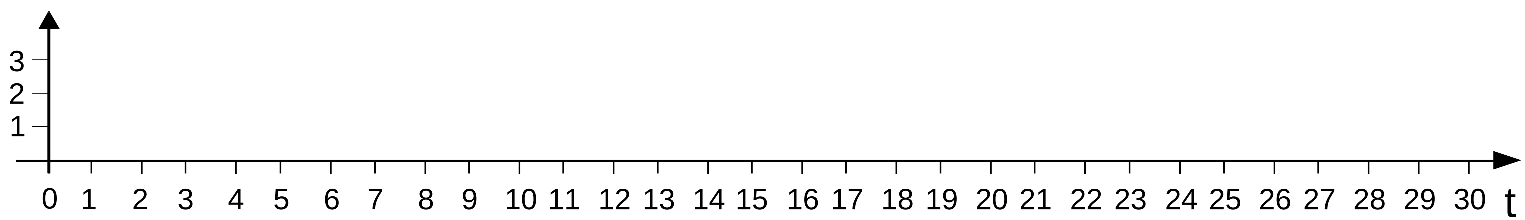
Waiting Queue Simulation – Step by Step

Simulation time	Current Event	Process Routine	Number of busy servers
Initialization	none	Insert first customer arrival; Insert simulation termination;	B(t) = 0

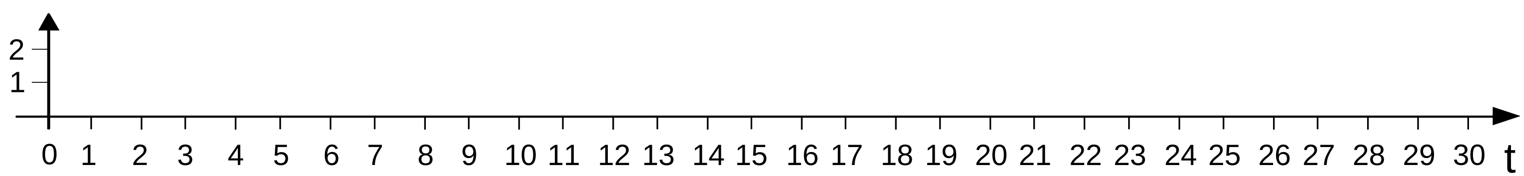
Event queue	Time of arrival of customers in queue	Number of waiting customers
[0;CA], [30;ST]		Q(t) = 0

Number of **arrived** customers = 0 Number of **served** customers = 0	Number of **blocked** customers = 0	$\sum_t Q(t) = 0$	$\sum_t B(t) = 0$

Q(t)



B(t)



Inter-arrival time										
Service completion time										

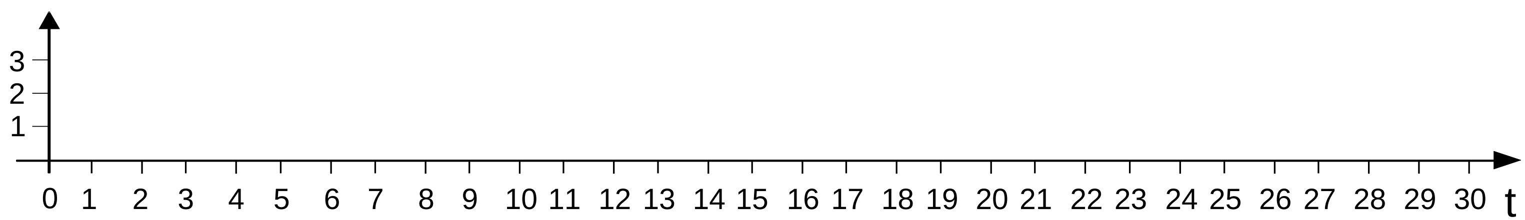
Waiting Queue Simulation – Step by Step

Simulation time	Current Event	Process Routine	Number of busy servers
0	Customer Arrival **[0;CA]**	Schedule next arrival->**IAT 2.0**; Increase B(t); Schedule service completion->**SCT 5.0**	B(t) = 1

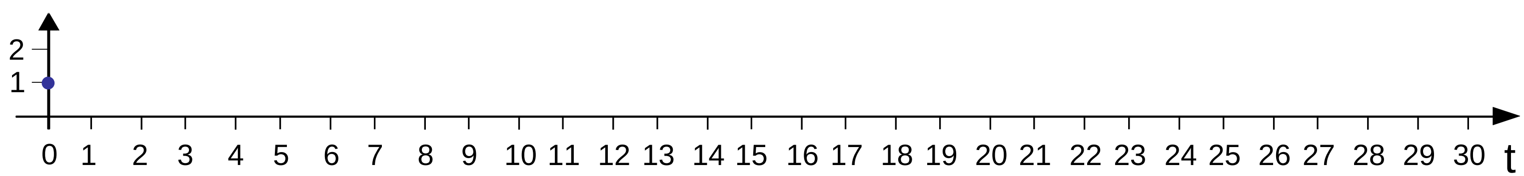
Event queue	Time of arrival of customers in queue	Number of waiting customers
[2;CA], **[5;SC]**, [30;ST]		Q(t) = 0

Number of **arrived** customers = 1 Number of **served** customers = 0	Number of **blocked** customers = 0	$\sum_t Q(t) = 0$	$\sum_t B(t) = 0$

Q(t)



B(t)



Inter-arrival time	**2.0**									
Service completion time	**5.0**									

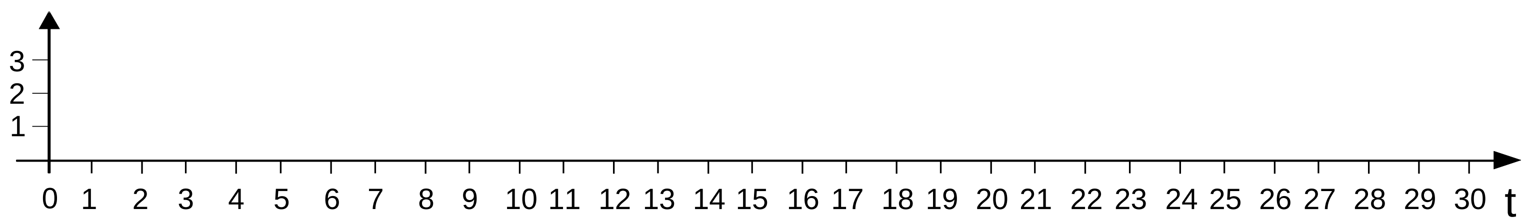
Waiting Queue Simulation – Step by Step

Simulation time	Current Event	Process Routine	Number of busy servers
2	Customer Arrival **[2;CA]**	Schedule next arrival->**IAT 1.0**; Increase B(t); Schedule service completion->**SCT 5.0**	B(t) = 2

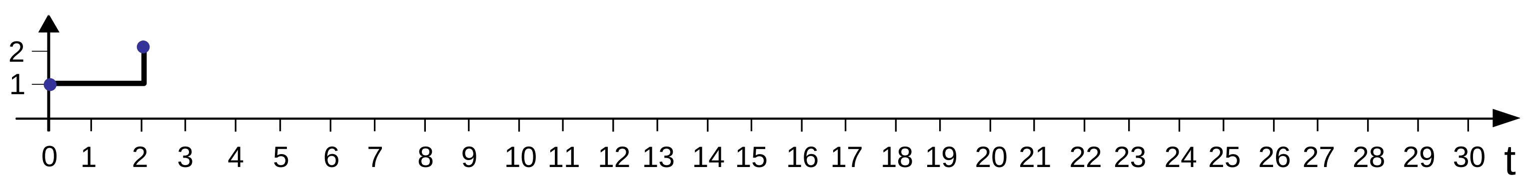
Event queue	Time of arrival of customers in queue	Number of waiting customers
[3;CA], [5;SC], **[7;SC],** [30;ST]		Q(t) = 0

Number of **arrived** customers = 2 Number of **served** customers = 0	Number of **blocked** customers = 0	$\sum_t Q(t) = 0$	$\sum_t B(t) = 2$

Q(t)



B(t)



Inter-arrival time	2.0	**1.0**								
Service completion time	5.0	**5.0**								

Waiting Queue Simulation – Step by Step

Simulation time	Current Event	Process Routine	Number of busy servers
3	Customer Arrival **[3;CA]**	Schedule next arrival->**IAT 1.0**; All servers busy->Insert customer in queue	B(t) = 2

Event queue	Time of arrival of customers in queue	Number of waiting customers
[4;CA], [5;SC], [7;SC], [30;ST]	**[3;CA]**	Q(t) = 1

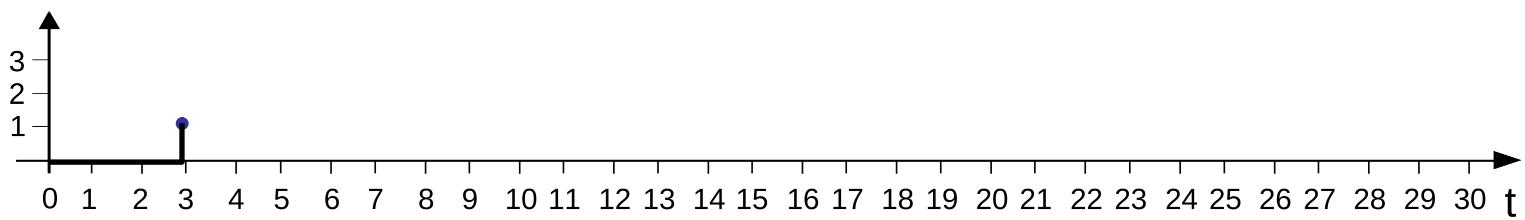
Number of **arrived** customers = 3
Number of **served** customers = 0

Number of **blocked** customers = 0

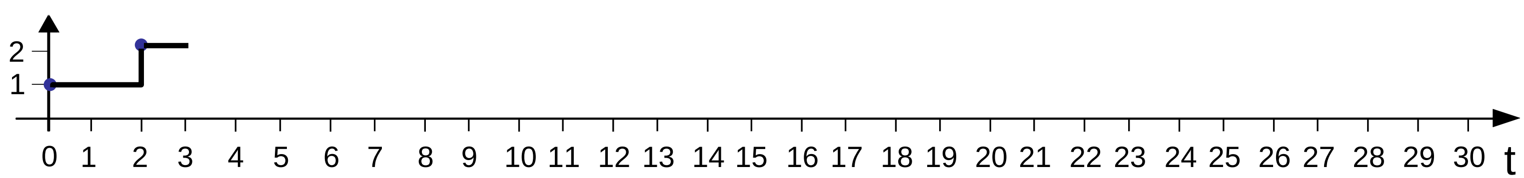
$\sum_t Q(t) = 0$

$\sum_t B(t) = 4$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	**1.0**							
Service completion time	5.0	5.0								

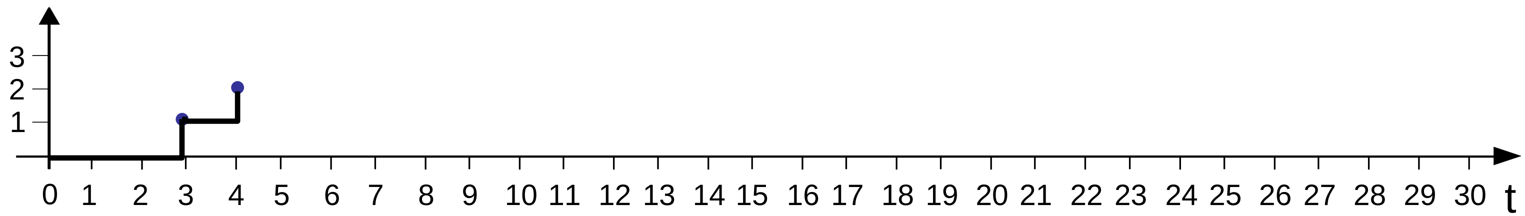
Waiting Queue Simulation – Step by Step

Simulation time	**Current Event**	**Process Routine**	**Number of busy servers**
4	Customer Arrival **[4;CA]**	Schedule next arrival->**IAT 2.0**; All servers busy->Insert customer in queue	B(t) = 2

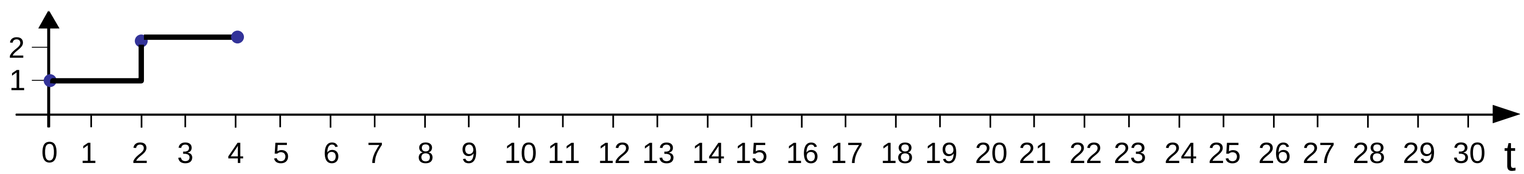
Event queue	Time of arrival of customers in queue	**Number of waiting customers**
[5;SC], **[6;CA],** [7;SC], [30;ST]	[3;CA], **[4;CA]**	Q(t) = 2

Number of **arrived** customers = 4 Number of **served** customers = 0	Number of **blocked** customers = 0	$\sum_t Q(t) = 1$	$\sum_t B(t) = 6$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	**2.0**						
Service completion time	5.0	5.0								

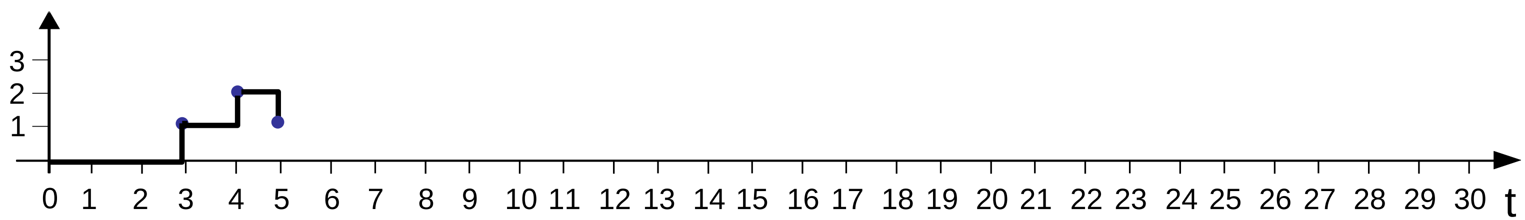
Waiting Queue Simulation – Step by Step

Simulation time	**Current Event**	**Process Routine**	**Number of busy servers**
5	Service Completion **[5;SC]**	Remove customer from queue; Schedule service completion -> **SCT 6.0**;	B(t) = 2

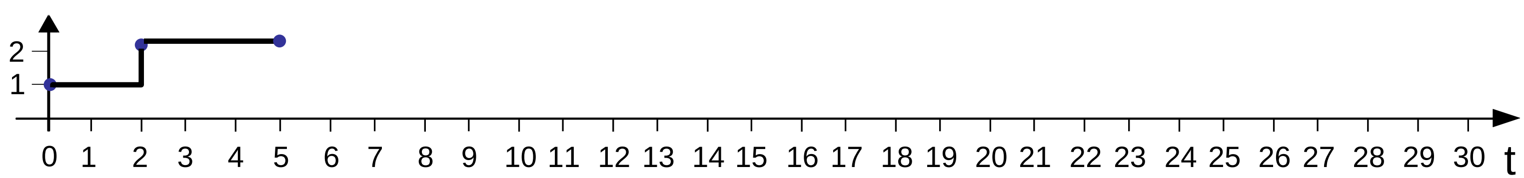
Event queue	Time of arrival of customers in queue	**Number of waiting customers**
[6;CA], [7;SC], **[11;SC],** [30;ST]	[4;CA]	Q(t) = 1

Number of **arrived** customers = 4 Number of **served** customers = 1	Number of blocked customers = 0	$\sum_t Q(t) = 3$	$\sum_t B(t) = 8$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	2.0						
Service completion time	5.0	5.0	**6.0**							

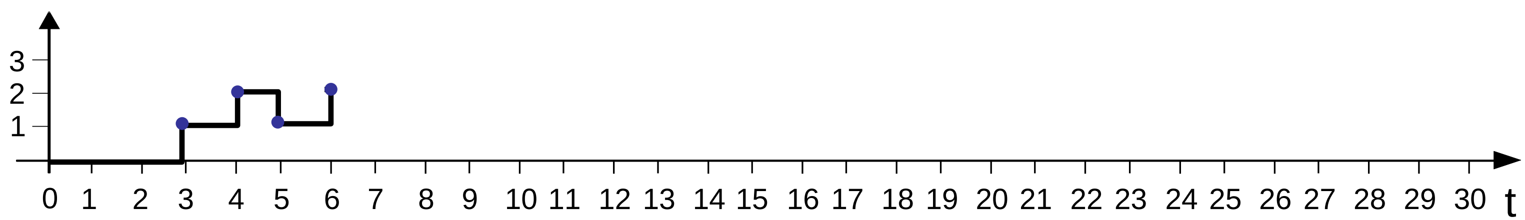
Waiting Queue Simulation – Step by Step

Simulation time	**Current Event**	**Process Routine**	**Number of busy servers**
6	Customer Arrival **[6;CA]**	Schedule next arrival->**IAT 1.0**; All servers busy->Insert customer in queue	B(t) = 2

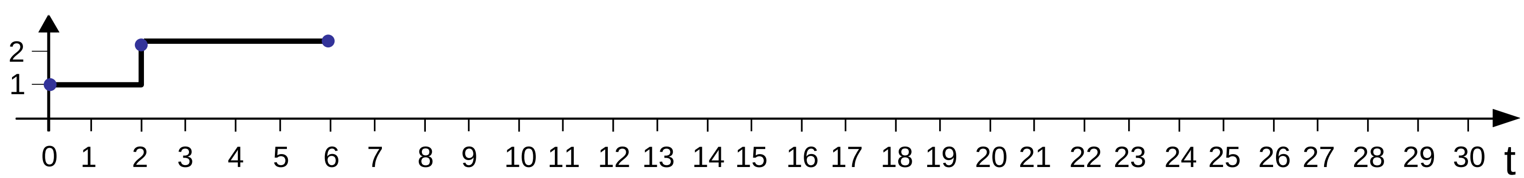
Event queue	Time of arrival of customers in queue	**Number of waiting customers**
[7;SC], **[7;CA],** [11;SC], [30;ST]	[4;CA], **[6;CA]**	Q(t) = 2

Number of **arrived** customers = 5 Number of **served** customers = 1	Number of blocked customers = 0	$\sum_t Q(t) = 4$	$\sum_t B(t) = 10$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	**1.0**					
Service completion time	5.0	5.0	6.0							

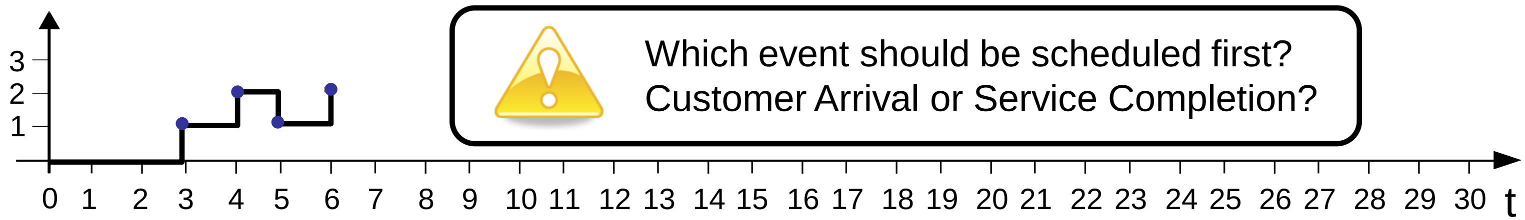
Waiting Queue Simulation – Step by Step

Simulation time	Current Event	Process Routine	Number of busy servers
7	Customer Arrival / Service Completion		B(t) = 2

Event queue	Time of arrival of customers in queue	Number of waiting customers
[7;SC], **[7;CA],** [11;SC], [30;ST]	[4;CA], [6;CA]	Q(t) = 2

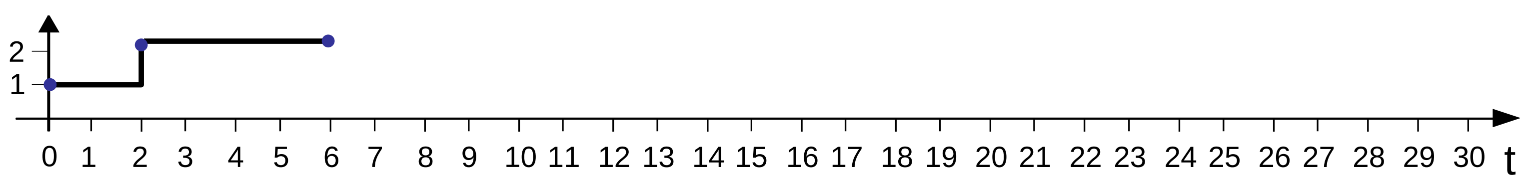
Number of **arrived** customers = X Number of **served** customers = X	Number of **blocked** customers = 0	$\sum_t Q(t) = 4$	$\sum_t B(t) = 10$

Q(t)



Which event should be scheduled first?
Customer Arrival or Service Completion?

B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	**1.0**					
Service completion time	5.0	5.0	6.0							

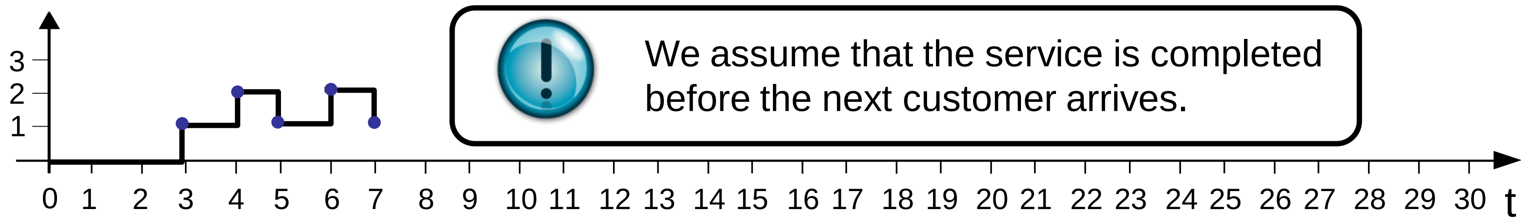
Waiting Queue Simulation – Step by Step

Simulation time	**Current Event**	**Process Routine**	**Number of busy servers**
7	Service Completion **[7;SC]**	Remove customer from queue; Schedule service completion -> **SCT 4.0**;	B(t) = 2

Event queue	Time of arrival of customers in queue	**Number of waiting customers**
[7;CA], [11;SC], [11;SC], [30;ST]	[6;CA]	Q(t) = 1

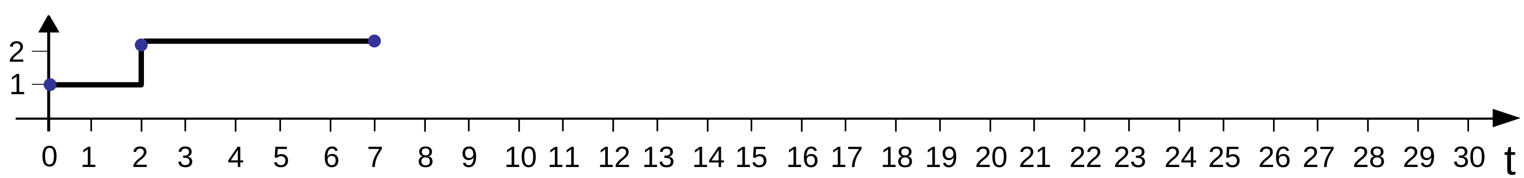
Number of **arrived** customers = 5 Number of **served** customers = 2	Number of **blocked** customers = 0	$\sum_t Q(t) = 6$	$\sum_t B(t) = 12$

Q(t)



We assume that the service is completed before the next customer arrives.

B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	1.0				
Service completion time	5.0	5.0	6.0	**4.0**					

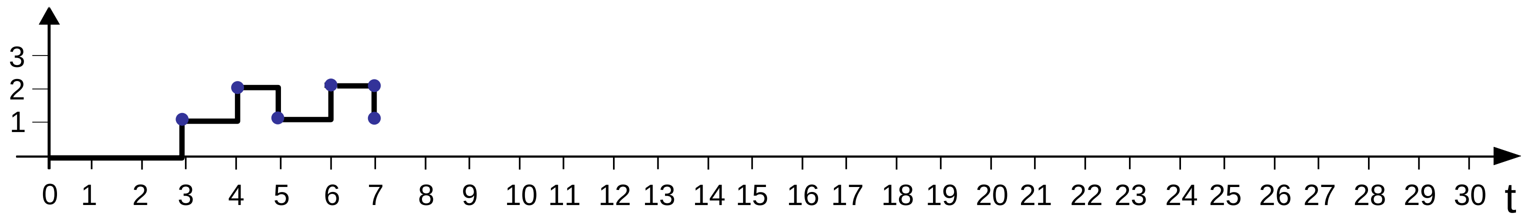
Waiting Queue Simulation – Step by Step

Simulation time	Current Event	Process Routine	Number of busy servers
7	Customer Arrival **[7;CA]**	Schedule next arrival->**IAT 2.0**; All servers busy->Insert customer in queue	B(t) = 2

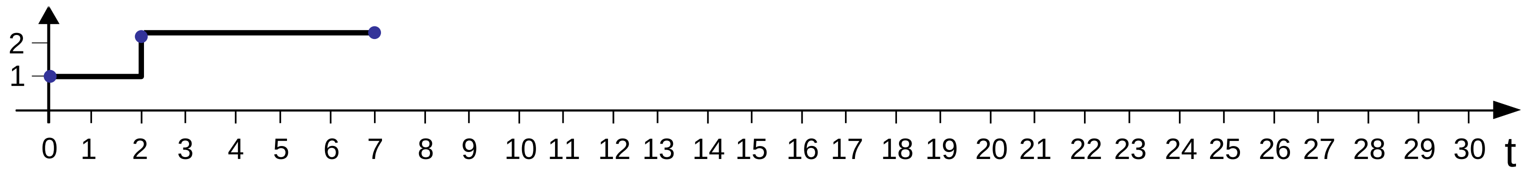
Event queue	Time of arrival of customers in queue	Number of waiting customers
[9;CA], [11;SC], [11;SC], [30;ST]	[6;CA], **[7;CA]**	Q(t) = 2

Number of **arrived** customers = 6 Number of **served** customers = 2	Number of **blocked** customers = 0	$\sum_t Q(t) = 6$	$\sum_t B(t) = 12$

Q(t)



B(t)

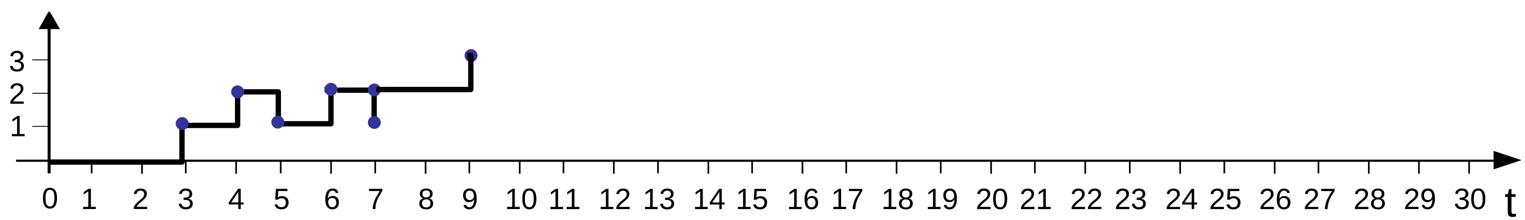


Inter-arrival time	2.0	1.0	1.0	2.0	1.0	**2.0**				
Service completion time	5.0	5.0	6.0	4.0						

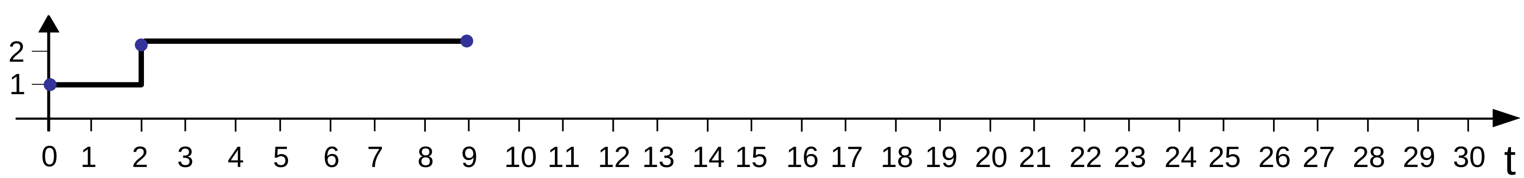
Waiting Queue Simulation – Step by Step

Simulation time 9	**Current Event** Customer Arrival **[9;CA]**	**Process Routine** Schedule next arrival->**IAT 1.0**; All servers busy->Insert customer in queue	**Number of busy servers** B(t) = 2
Event queue **[10;CA],** [11;SC], [11;SC], [30;ST]	Time of arrival of customers in queue [6;CA], [7;CA], **[9;CA]**	**Number of waiting customers** Q(t) = 3	
Number of **arrived** customers = 7 Number of **served** customers = 2	Number of **blocked** customers = 0	$\sum_t Q(t) = 10$	$\sum_t B(t) = 16$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	1.0	2.0	**1.0**			
Service completion time	5.0	5.0	6.0	4.0						

Waiting Queue Simulation – Step by Step

Simulation time	Current Event	Process Routine	Number of busy servers
10	Customer Arrival **[10;CA]**	Schedule next arrival->**IAT 4.0**; All servers busy->Queue full ->Block customer	B(t) = 2

Event queue

[11;SC], [11;SC], [14;CA], [30;ST]

Time of arrival of customers in queue

[6;CA], [7;CA], [9;CA]

Number of waiting customers

Q(t) = 3

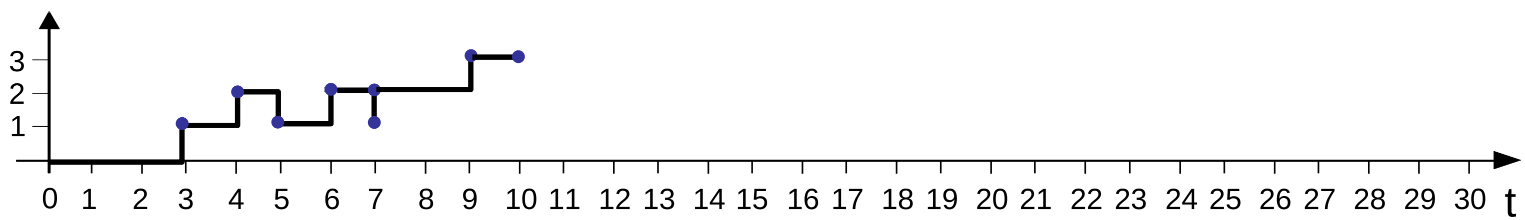
Number of **arrived** customers = 8
Number of **served** customers = 2

Number of **blocked** customers = 1

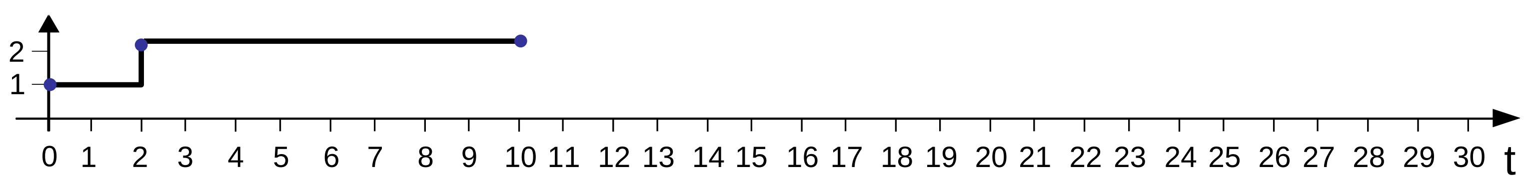
$$\sum_t Q(t) = 13$$

$$\sum_t B(t) = 18$$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	1.0	2.0	1.0	4.0		
Service completion time	5.0	5.0	6.0	4.0						

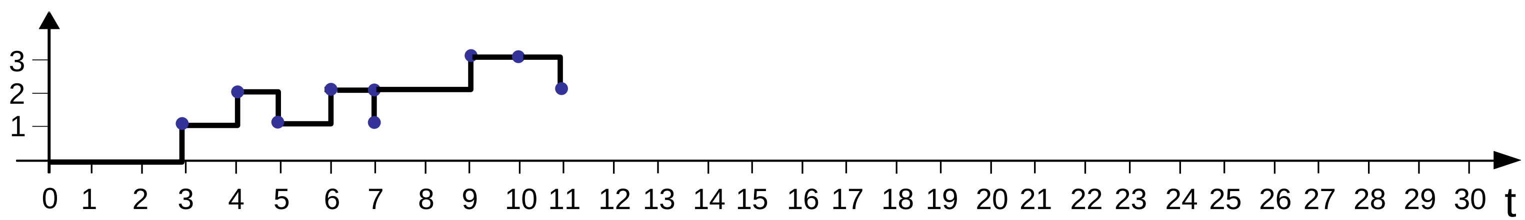
Waiting Queue Simulation – Step by Step

Simulation time	Current Event	Process Routine	Number of busy servers
11	Service Completion **[11;SC]**	Remove customer from queue; Schedule service completion -> **SCT 4.0**;	B(t) = 2

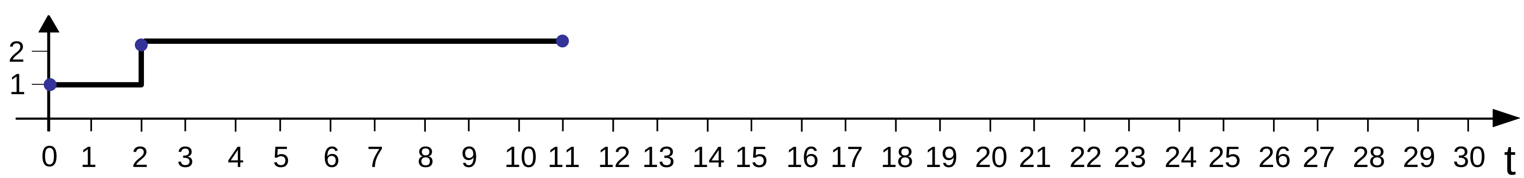
Event queue	Time of arrival of customers in queue	Number of waiting customers
[15;SC], [14;CA], [30;ST]	[7;CA], [9;CA]	Q(t) = 2

Number of **arrived** customers = 8 Number of **served** customers = 3	Number of **blocked** customers = 1	$\sum_t Q(t) = 16$	$\sum_t B(t) = 20$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	1.0	2.0	1.0	4.0		
Service completion time	5.0	5.0	6.0	**4.0**						

Waiting Queue Simulation – Step by Step

Simulation time	Current Event	Process Routine	Number of busy servers
11	Service Completion **[11;SC]**	Remove customer from queue; Schedule service completion -> **SCT 6.0**;	B(t) = 2

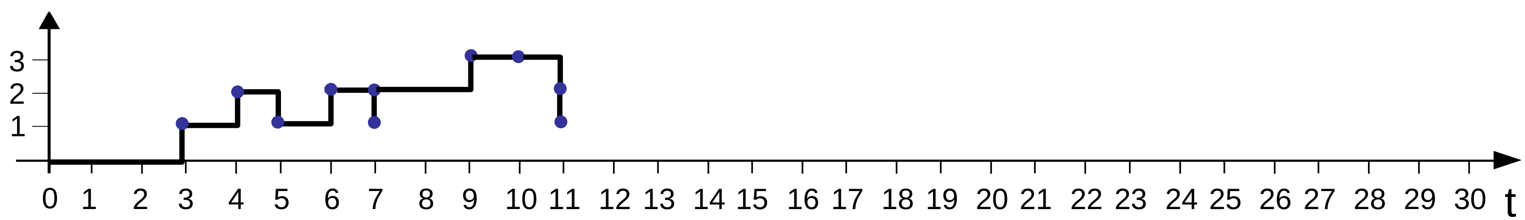
Event queue	Time of arrival of customers in queue	Number of waiting customers
[14;CA], [15;SC], **[17;SC],** [30;ST]	[9;CA]	Q(t) = 1

Number of **arrived** customers = 8
Number of **served** customers = 4

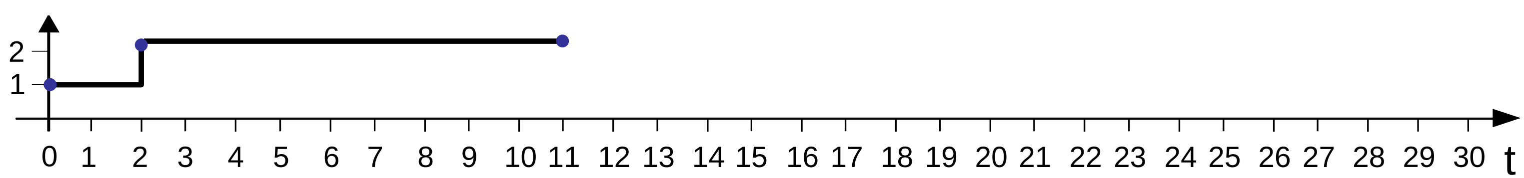
Number of **blocked** customers = 1

$\sum_t Q(t) = 16$ $\sum_t B(t) = 20$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	1.0	2.0	1.0	4.0		
Service completion time	5.0	5.0	6.0	4.0	**6.0**					

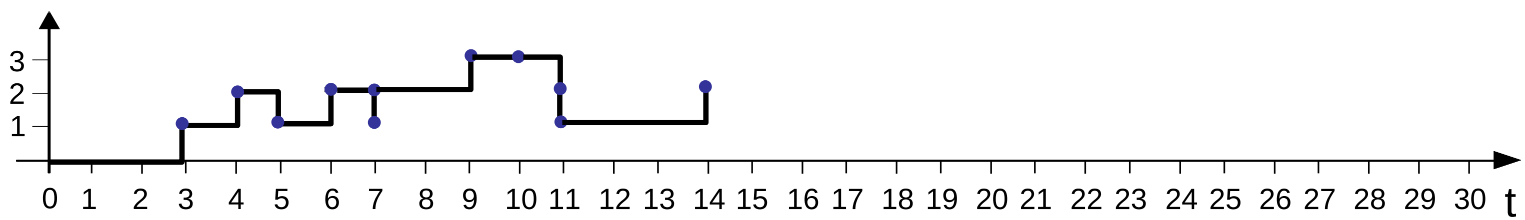
Waiting Queue Simulation – Step by Step

Simulation time	Current Event	Process Routine	Number of busy servers
14	Customer Arrival **[14;CA]**	Schedule next arrival->**IAT 8.0**; All servers busy->Insert customer in queue	B(t) = 2

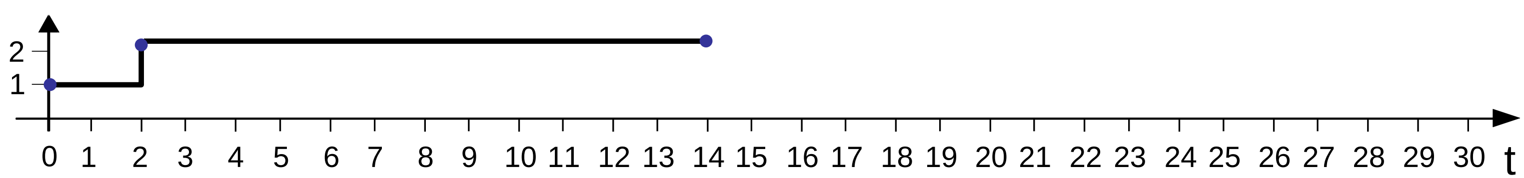
Event queue	Time of arrival of customers in queue	**Number of waiting customers**
[15;SC], [17;SC], **[22;CA]**, [30;ST]	[9;CA], [14;CA]	Q(t) = 2

Number of **arrived** customers = 9 Number of **served** customers = 4	Number of **blocked** customers = 1	$\sum_t Q(t) = 19$	$\sum_t B(t) = 26$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	1.0	2.0	1.0	4.0	**8.0**	
Service completion time	5.0	5.0	6.0	4.0	6.0					

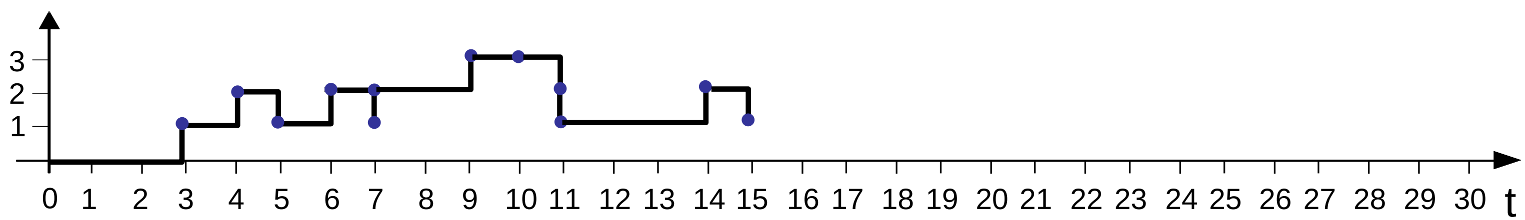
Waiting Queue Simulation – Step by Step

Simulation time	**Current Event**	**Process Routine**	**Number of busy servers**
15	Service Completion **[15;SC]**	Remove customer from queue; Schedule service completion -> **SCT 3.0**;	B(t) = 2

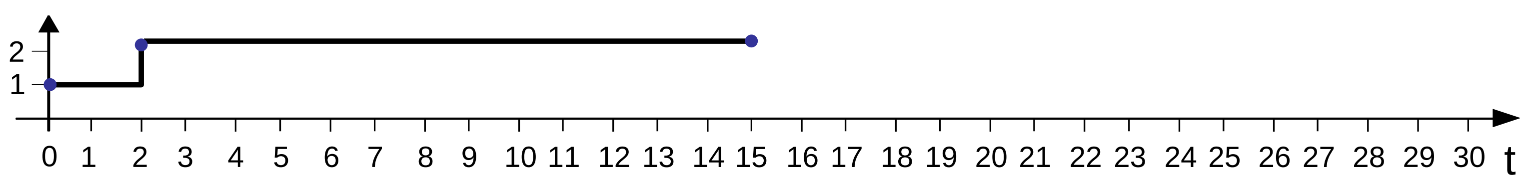
Event queue	Time of arrival of customers in queue	**Number of waiting customers**
[17;SC], **[18;SC],** [22;CA], [30;ST]	[14;CA]	Q(t) = 1

Number of **arrived** customers = 9 Number of **served** customers = 5	Number of **blocked** customers = 1	$\sum_t Q(t) = 21$	$\sum_t B(t) = 28$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	1.0	2.0	1.0	4.0	8.0	
Service completion time	5.0	5.0	6.0	4.0	6.0	**3.0**				

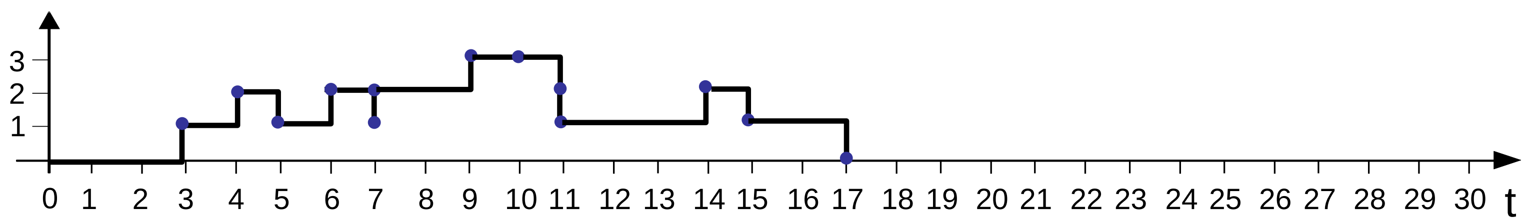
Waiting Queue Simulation – Step by Step

Simulation time	**Current Event**	**Process Routine**	**Number of busy servers**
17	Service Completion **[17;SC]**	Remove customer from queue; Schedule service completion -> **SCT 3.0**;	B(t) = 2

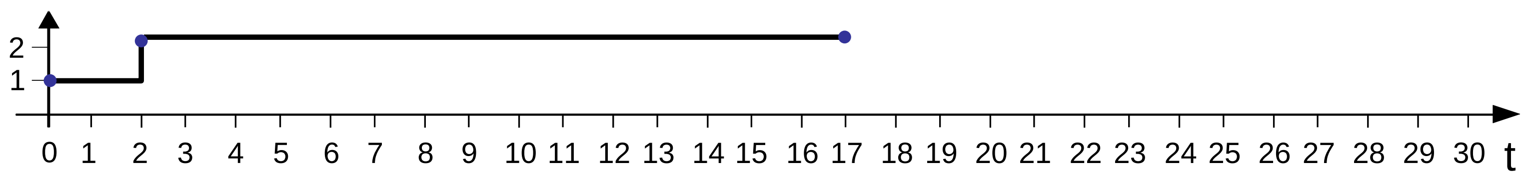
Event queue	Time of arrival of customers in queue	**Number of waiting customers**
[18;SC], [20;SC], [22;CA], [30;ST]		Q(t) = 0

Number of **arrived** customers = 9 Number of **served** customers = 6	Number of **blocked** customers = 1	$\sum_t Q(t) = 23$	$\sum_t B(t) = 32$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	1.0	2.0	1.0	4.0	8.0	
Service completion time	5.0	5.0	6.0	4.0	6.0	3.0	**3.0**			

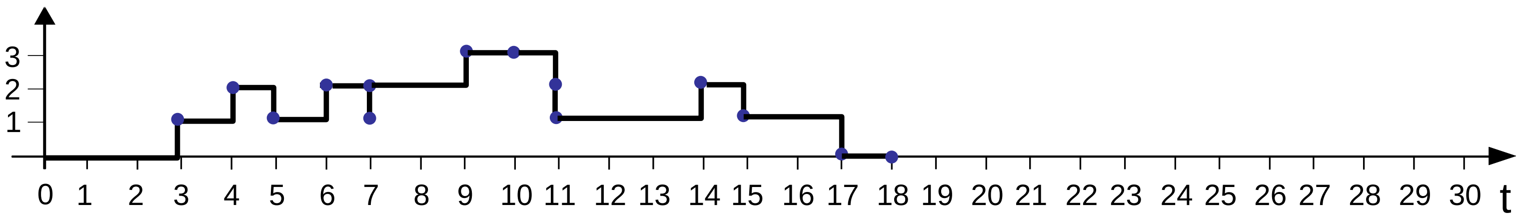
Waiting Queue Simulation – Step by Step

Simulation time	**Current Event**	**Process Routine**	**Number of busy servers**
18	Service Completion **[18;SC]**	Set one server to idle	B(t) = 1

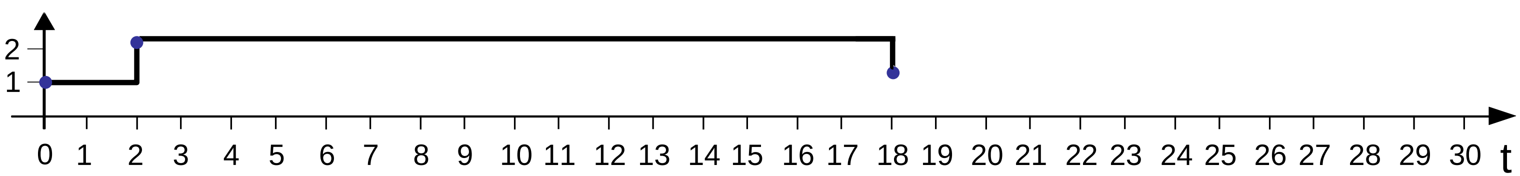
Event queue	Time of arrival of customers in queue	**Number of waiting customers**
[20;SC], [22;CA], [30;ST]		Q(t) = 0

Number of **arrived** customers = 9 Number of **served** customers = 7	Number of **blocked** customers = 1	$\sum_t Q(t) = 23$	$\sum_t B(t) = 34$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	1.0	2.0	1.0	4.0	8.0	
Service completion time	5.0	5.0	6.0	4.0	6.0	3.0	3.0			

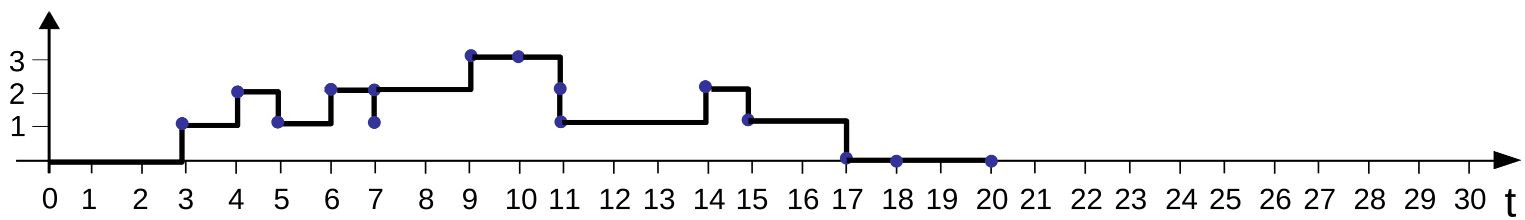
Waiting Queue Simulation – Step by Step

Simulation time	**Current Event**	**Process Routine**	**Number of busy servers**
20	Service Completion **[20;SC]**	Set server to idle	B(t) = 0

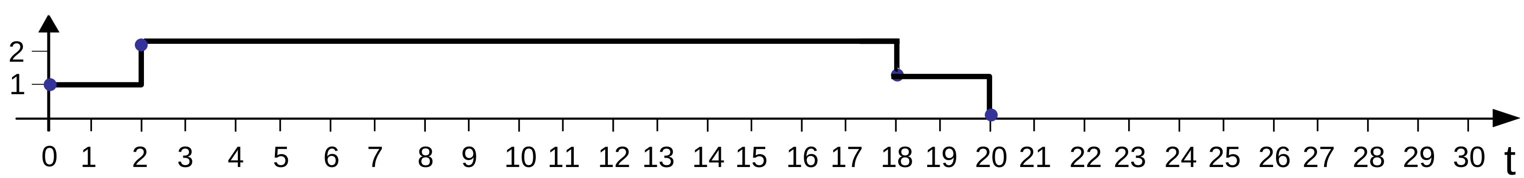
Event queue	Time of arrival of customers in queue	**Number of waiting customers**
[22;CA], [30;ST]		Q(t) = 0

Number of **arrived** customers = 9 Number of **served** customers = 8	Number of **blocked** customers = 1	$\sum_t Q(t) = 23$	$\sum_t B(t) = 36$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	1.0	2.0	1.0	4.0	8.0	
Service completion time	5.0	5.0	6.0	4.0	6.0	3.0	3.0			

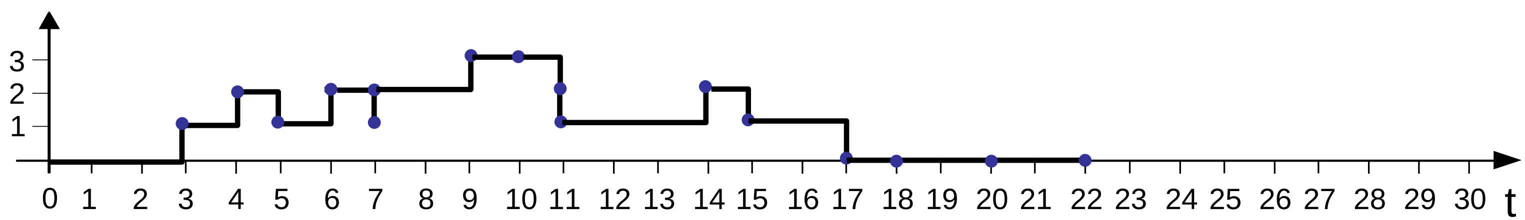
Waiting Queue Simulation – Step by Step

Simulation time	**Current Event**	**Process Routine**	**Number of busy servers**
22	Customer Arrival **[22;CA]**	Schedule next arrival->**IAT 6.0**; Set server to busy; Schedule service completion-> **SCT 9.0**;	B(t) = 1

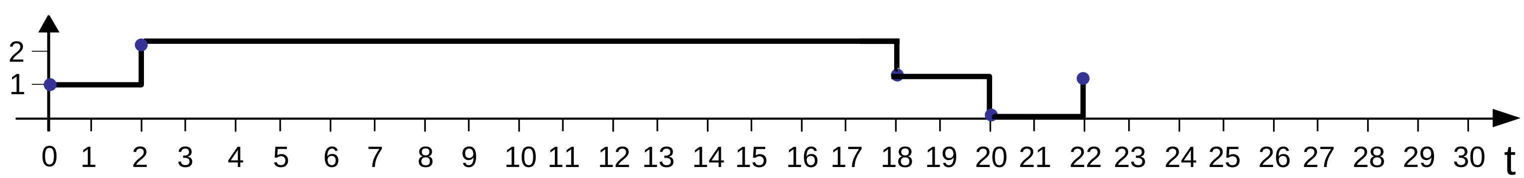
Event queue	Time of arrival of customers in queue	**Number of waiting customers**
[28;CA], [30;ST], **[31;SC]**		Q(t) = 0

Number of **arrived** customers = 10 Number of **served** customers = 8	Number of **blocked** customers = 1	$\sum_t Q(t) = 23$	$\sum_t B(t) = 36$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	1.0	2.0	1.0	4.0	8.0	**6.0**
Service completion time	5.0	5.0	6.0	4.0	6.0	3.0	3.0	9.0		

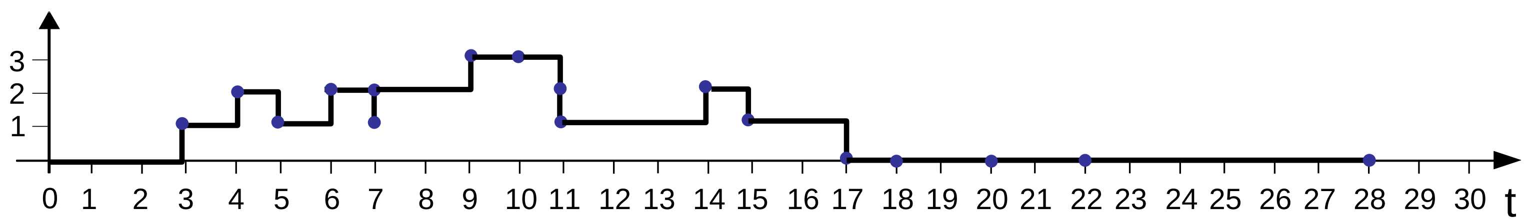
Waiting Queue Simulation – Step by Step

Simulation time	Current Event	Process Routine	Number of busy servers
28	Customer Arrival **[28;CA]**	Schedule next arrival->**IAT 3.0**; Set server to busy; Schedule service completion-> **SCT 4.0**;	B(t) = 2

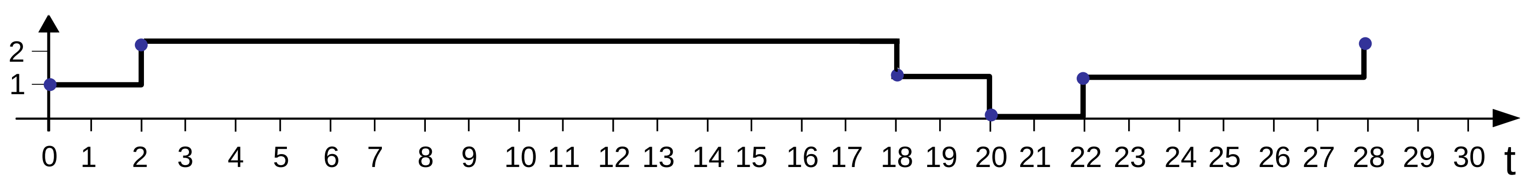
Event queue	Time of arrival of customers in queue	Number of waiting customers
[30;ST], [31;SC] , **[31;CA], [32;SC]**		Q(t) = 0

Number of **arrived** customers = 11 Number of **served** customers = 8	Number of **blocked** customers = 1	$\sum_t Q(t) = 23$	$\sum_t B(t) = 42$

Q(t)



B(t)

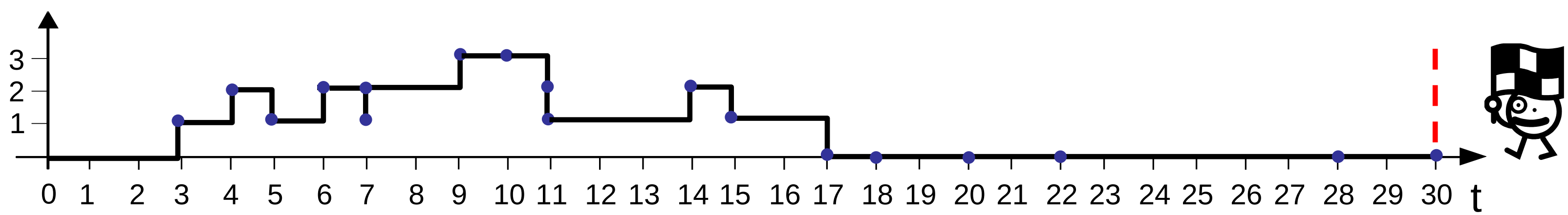


Inter-arrival time	2.0	1.0	1.0	2.0	1.0	2.0	1.0	4.0	8.0	6.0
Service completion time	5.0	5.0	6.0	4.0	6.0	3.0	3.0	9.0	**4.0**	

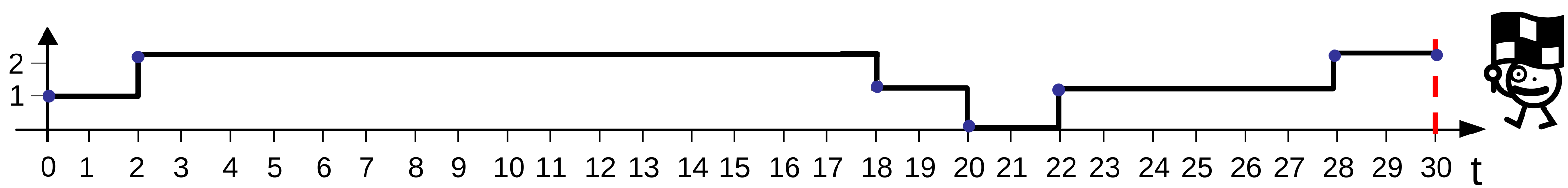
Waiting Queue Simulation – Step by Step

Simulation time	Current Event	Process Routine	Number of busy servers
30	Simulation Termination **[30;ST]**	End simulation->exit while loop	B(t) = 2
Event queue [31;SC] , [31;CA], [32;SC]	Time of arrival of customers in queue		**Number of waiting customers** Q(t) = 0
Number of **arrived** customers = 11 Number of **served** customers = 8	Number of **blocked** customers = 1	$\sum_t Q(t) = 23$	$\sum_t B(t) = 46$

Q(t)



B(t)



Inter-arrival time	2.0	1.0	1.0	2.0	1.0	2.0	1.0	4.0	8.0	6.0
Service completion time	5.0	5.0	6.0	4.0	6.0	3.0	3.0	9.0	4.0	

Waiting Queue Simulation – Step by Step

- Statistics:
  - Simulation duration: = 30 ticks
  - Number of **arrived** customers = 11
  - Number of **served** customers = 8
  - Number of **blocked** customers = 1
  - Sum of waiting time $\sum_t Q(t) = 23$
  - Sum of server utilization $\sum_t B(t) = 46$
  - Inter-arrival times 2.0;1.0;1.0;2.0;1.0;2.0;1.0;4.0;8.0;6.0
  - Service-completion times 5.0;5.0;6.0;4.0;6.0;3.0;3.0;9.0;4.0

Only executed events are considered

Waiting Queue Simulation – Step by Step

- Statistics:
  - Average waiting time $= \dfrac{\text{Sum of waiting time}}{\text{Number of customer arrivals}} = \dfrac{23}{11}$
  - Average server utilization $= \dfrac{\text{Sum of server utilization}}{\text{Simulation duration * \#Servers}} = \dfrac{46}{2*30}$
  - Average Inter-arrival time $= \dfrac{\text{Sum of IAT}}{\text{Number of customer arrivals - 1}} = \dfrac{28}{10}$
  - Service-completion times $= \dfrac{\text{Sum of SCT}}{\text{Number of service completions}} = \dfrac{45}{9}$