Simulation of Computer Networks

Holger Füßler

Lehrstuhl für Praktische Informatik IV, University of Mannheim

Vorbemerkungen

- **Lehrstuhl für Praktische Informatik IV**
  - **http://www.informatik.uni-mannheim.de/informatik/pi4**
- **Seit August 2001**
- **Gegenwärtige Arbeitsgebiete:**
  - **Wireless ad hoc networks, insbesondere: vehicular ad hoc networks**

Simulation von Rechnernetzen

- **Vertieft Vorlesungen ‘Rechnernetze’**
- **Widmet sich den Fragen:**
  - **Wie kann ich Protokolle ausprobieren?**
  - **Wie kann ich verschiedene Netzkonfigurationen quantitativ vergleichen (ohne die Netze zu bauen)?**
- **Soll Hilfestellung für Studien-/Diplomarbeiten**
  - **In vielen Arbeiten wird simuliert.**

Prüfungsregelung

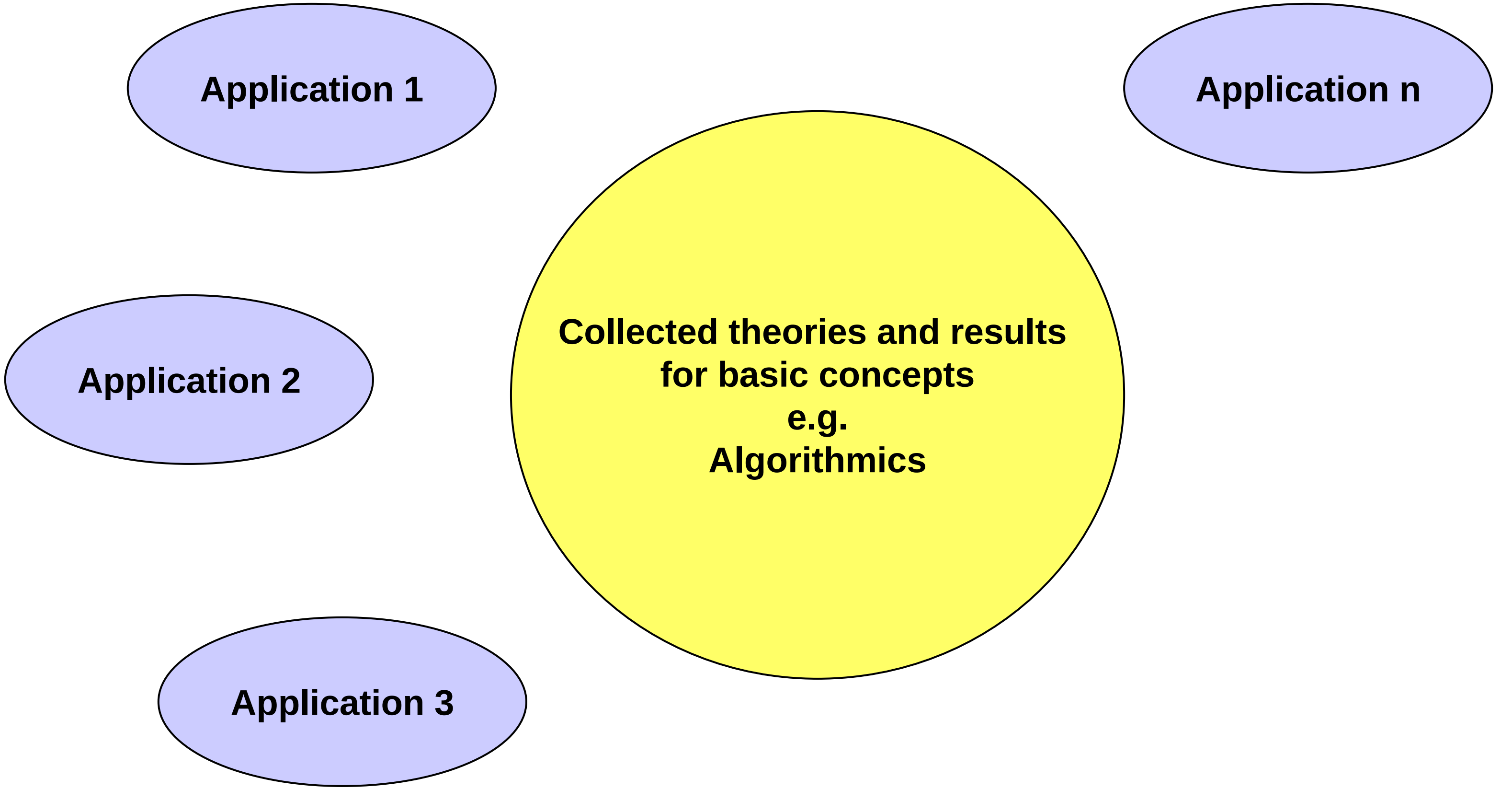
- **3 ECTS Punkte**
- **mündliche oder schriftliche Prüfung (je nach Nachfrage)**

Sprechzeiten, Vorlesungsfolien

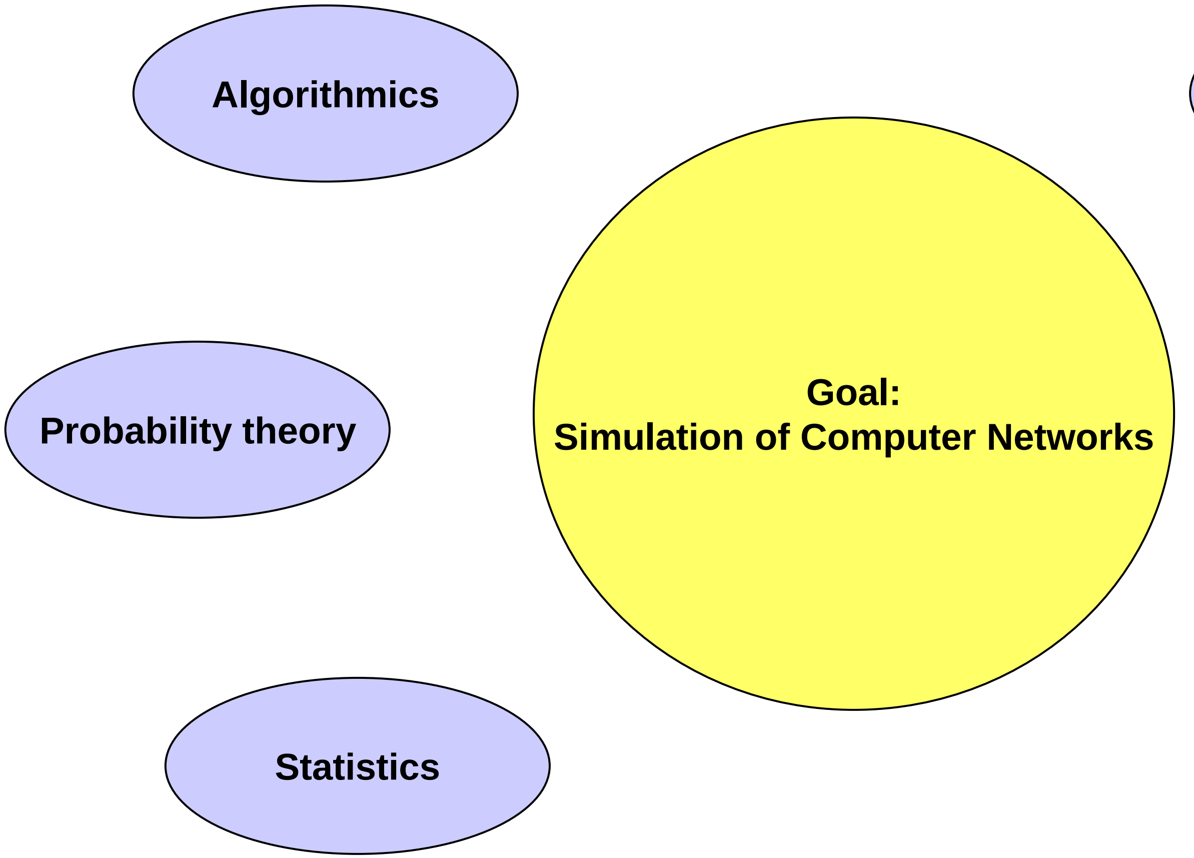
- **Holger Füßler (Sprechstunde nach Vereinbarung)**
  - **Am besten per e-mail Termin vereinbaren (fuessler@informatik.uni-mannheim.de)**

- **Die Vorlesungsfolien finden sich unter**
  - **http://www.informatik.uni-mannheim.de/informatik/pi4/stud/veranstaltungen/SS2004/netsim**
  - **geschützter Bereich**
  - **User: studi**
    pwd: simmel

Some lectures are like this …



NetSim is like this …



… requires broad knowledge on various fields

Prerequisites/Literature

- Basics (Grundstudium) in CS / Math / Statistics
- Networking: Rechnernetze

- Averill M. Law, W. David Kelton: “Simulation Modeling and Analysis”, McGraw-Hill, 3rd edition, 2000.
- Sheldon M. Ross: “Simulation”, 2nd edition, Academic Press, 1997.
- Stochastics, statistics: Anderson et al: “Schätzen und Testen”
- Computer networks: Andrew S. Tanenbaum: “Computer Networks”
- Pointers to original work is given on a ‘per lecture basis’.

Start of NetSim

Overview of first lecture

- **Part I: An 'abstract' view to simulations (top-down)**
  - **Simulation as *one* strategy to study a system**
  - **The big picture**

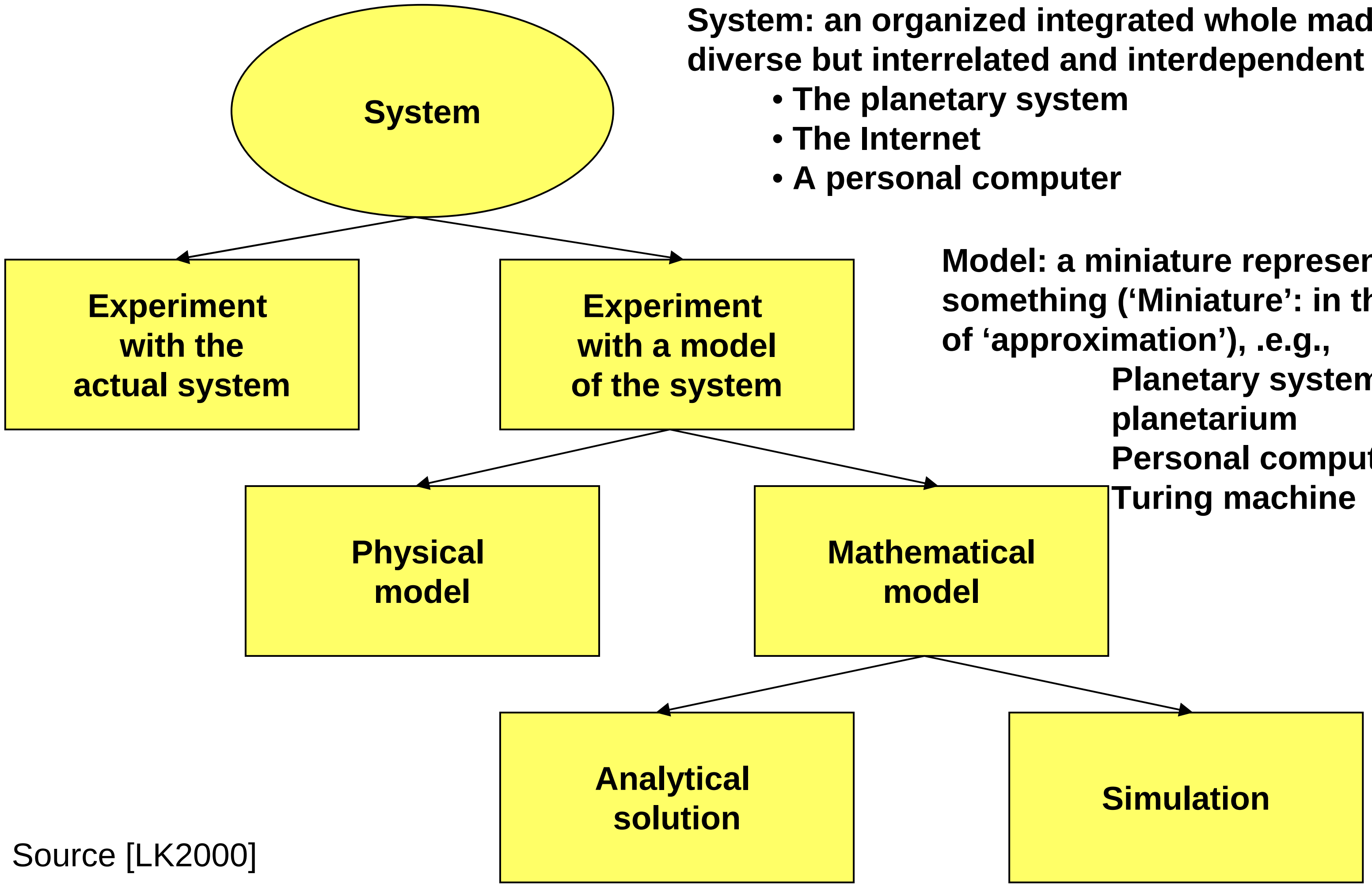
- **Part II: A 'concrete' simulation example (bottom-up)**

- **Part III: Course overview**
  - **Elements needed for simulation of computer networks**

I Ways to study a system

System: an organized integrated whole made up of diverse but interrelated and interdependent parts

- The planetary system
- The Internet
- A personal computer



Model: a miniature representation of something (‘Miniature’: in the sense of ‘approximation’), .e.g.,

Planetary system – planetarium

Personal computer – Turing machine

Source [LK2000]

I Simulation: advantages

» **Experiment with the actual system: too expensive, sometimes impossible (e.g., system does not exist yet)**
  – **Simulation is relatively inexpensive**
  – **Simulation works for concepts and ideas**

» **Experiment with a physical model: still expensive, needs a lot of work, some things cannot be ‚miniaturized‘ (e.g., radio propagation characteristics)**
  – **Simulation is cost-effective**
  – **Simulation allows for various degrees of accuracy**

» **Analytical treatment: most times models are too complex**
  – **Simulation allows for observation of the models behavior over time**

I Example: vehicular ad hoc networks

System

Experiment with the actual system

Experiment with a model of the system

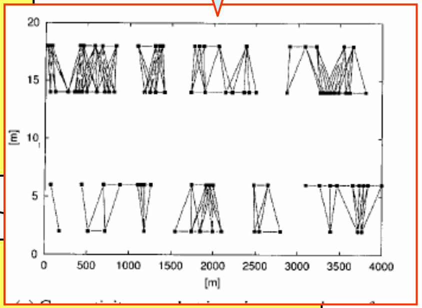
Physical model

Mathematical model

Analytical solution

Simulation

Models:
- Movement pattern
- Communication protocols
- Data traffic
- and their interactions

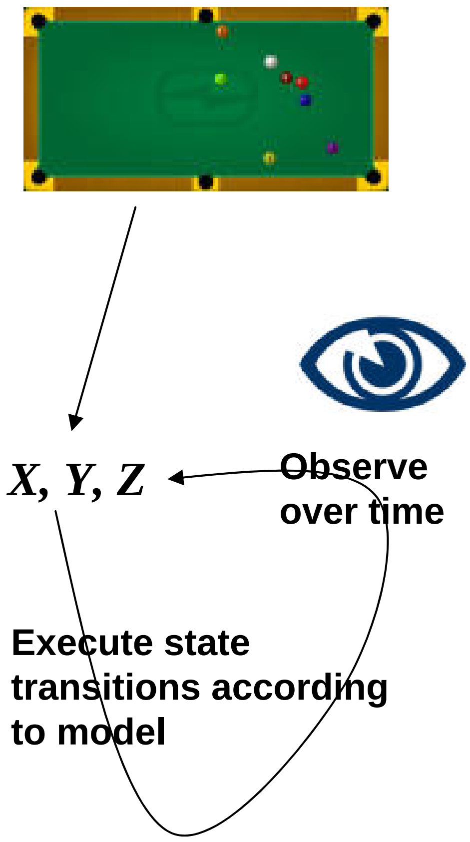


$$E\{C\} = \frac{1}{\alpha^2\beta^2}\sum_{\xi_1=1}^{\alpha}\sum_{\psi_1=1}^{\beta}\sum_{\xi_2=1}^{\alpha}\sum_{\psi_2=1}^{\beta} c(\xi_1, \psi_1, \xi_2, \psi_2)\,.$$

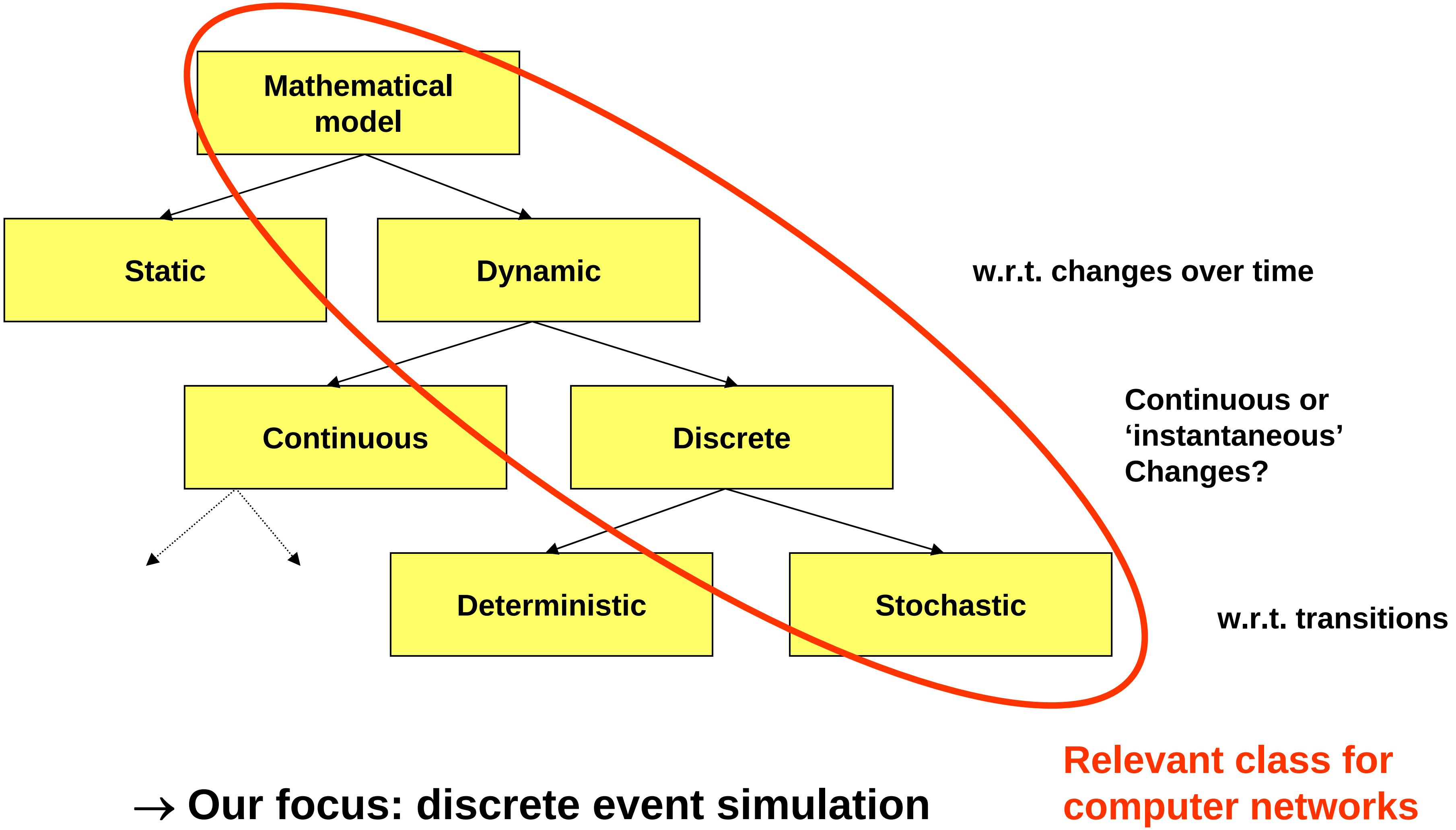
I The nature of simulation

- **Systems, system state, and system state changes**
  - We define the *state* of a system to be that collection of variables necessary to describe a system at a particular time, relative to the objectives of the study
  - In dynamic systems, the state of the systems changes over time
  - Usually, the local behavior of the system is known but the ‚evolution‘ of the system on a global scale is unknown.

- **Simulation**
  - Step 1: build a (virtual) model w.r.t. system states and their corresponding state transitions
  - Step 2: execute the model, i.e., the transition rules, and observe the output

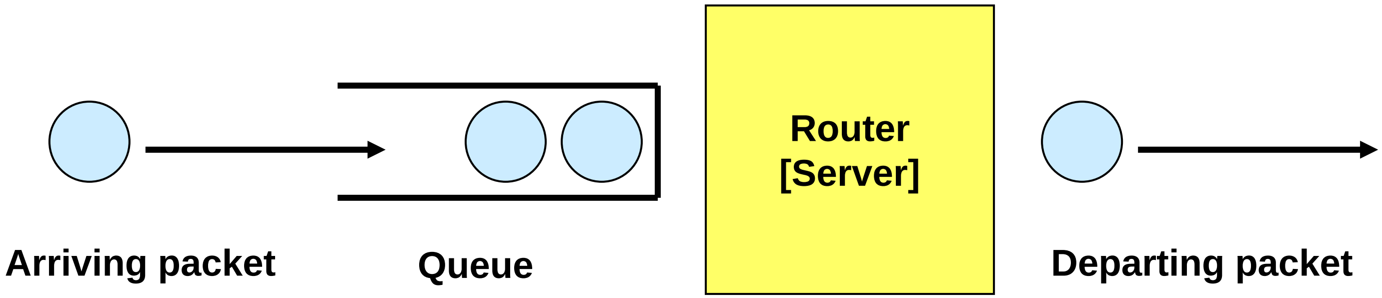


I Classification of models and simulation types



→ **Our focus: discrete event simulation**

II Classical introductory example: M/M/1 queue



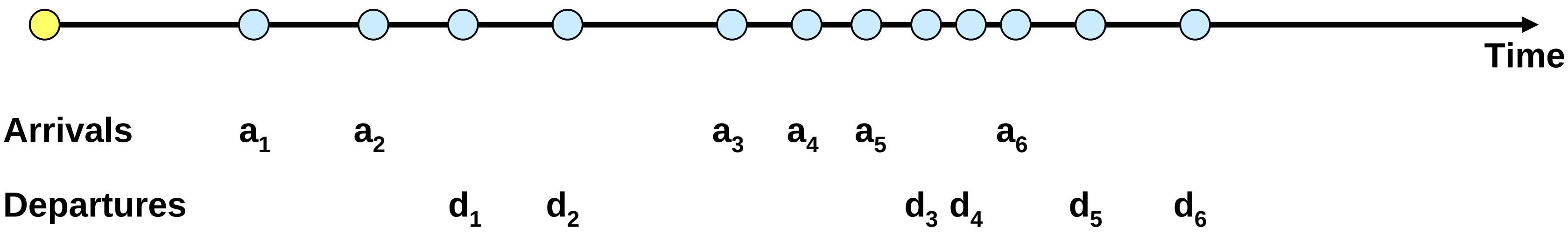
- Queuing systems as delay models
- Arrival process: ‘M’ for ‘memoryless’ (thus, exponentially distributed inter-arrival times)
- Service process: ‘M’ for ‘memoryless’ (thus, exponentially distributed service times)
- Number of queuing stations: 1

β=**1.0 s for inter-arrival times**

β=**0.5 s for service times**

$$f(x) = \frac{1}{\beta} e^{-x/\beta}$$

II Introductory example: next-event time advance



- **Events:**
  - Packet arrivals
  - Departure: depends on arrival, delay, and service time
- **Next-event time advance mechanism:**
  - Simulation clock advances to next event
    - State of system is updated
    - Knowledge of the times of occurrence of future events is updated
    - Go to next event
  - Thus, periods of inactivity are ‚skipped'.

II Introductory example: performance measures

Statistics for performance measures:

- **Average packet delay in queue:**
  - **Assume n packets are sent**
  - **Denote the delay of packet i by D_i**

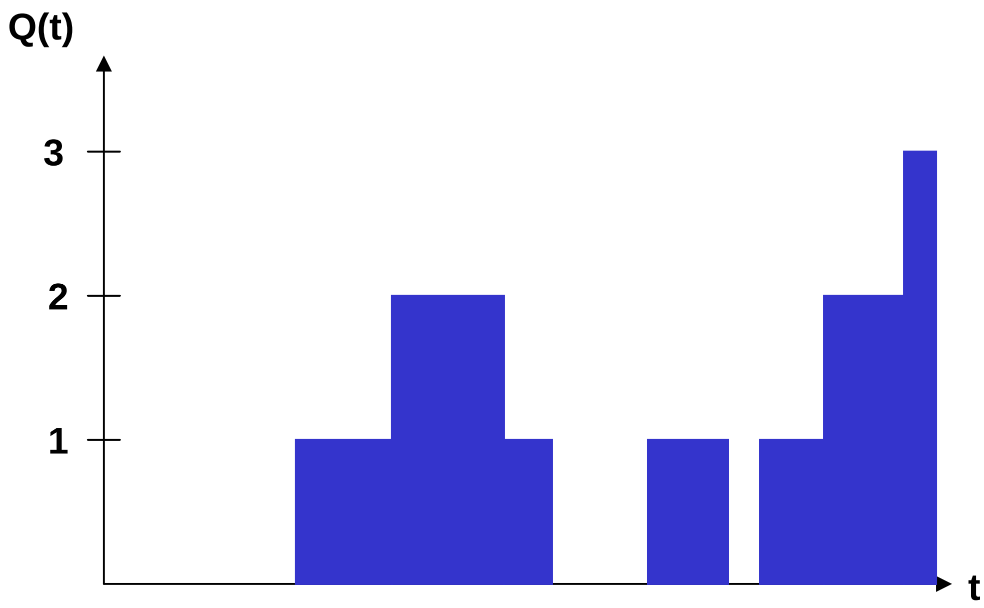
$$\hat{d}(n) = \frac{\sum_{i=1}^{n} D_i}{n}$$

Estimator or ‘statistic’

II Introductory example: performance measures

» **Time-average number of packets in queue**

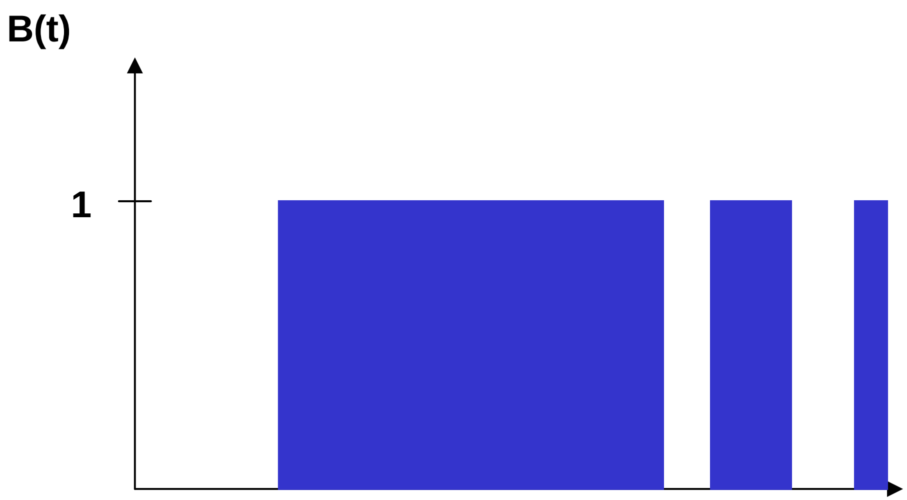
- **Let Q(t) denote the number of packets in the queue at time t**
- **Let T(n) denote the total simulation time for n packets.**



$$\hat{q}(n) = \frac{\int_0^{T(n)} Q(t)dt}{T(n)}$$

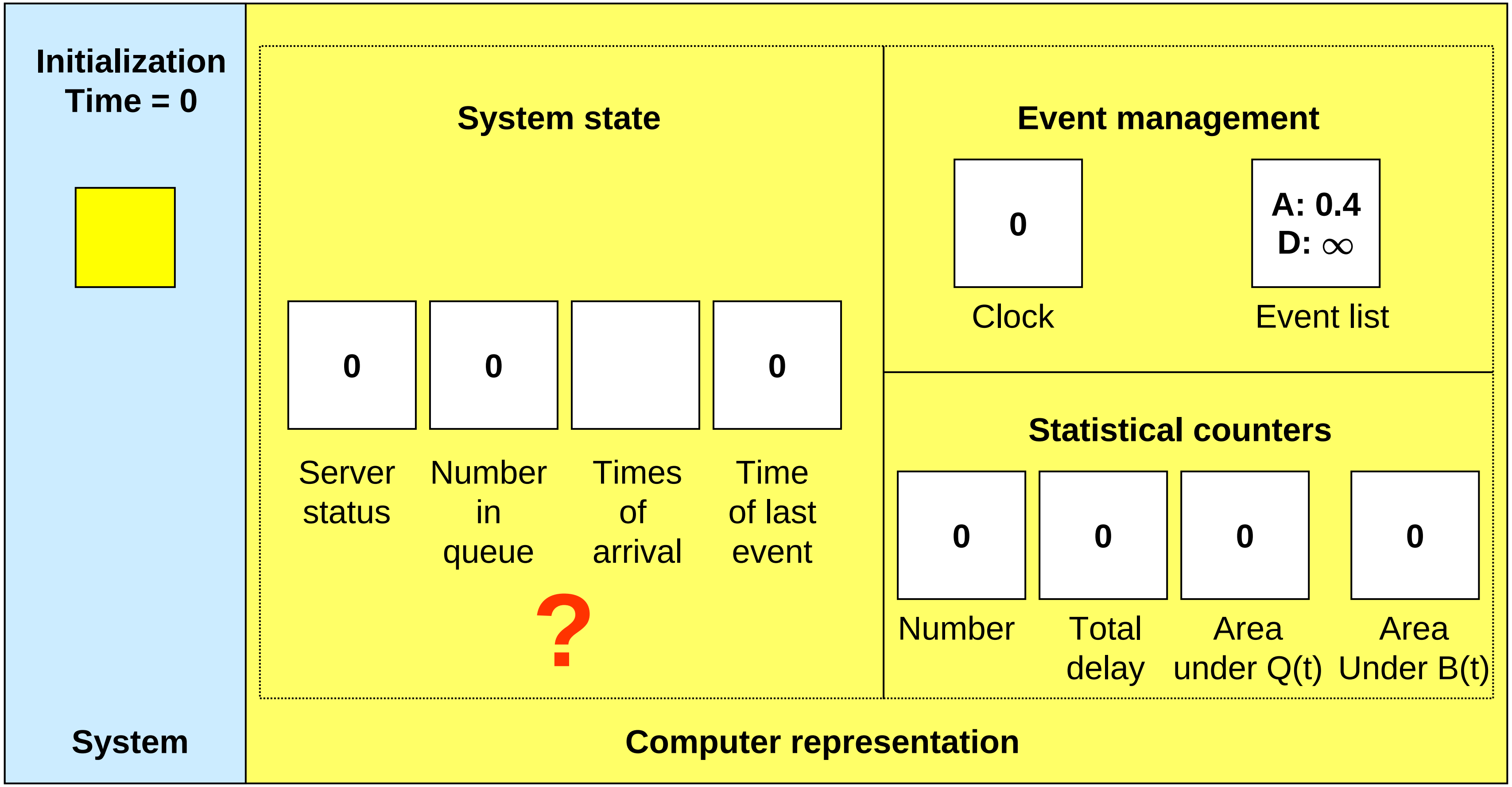
II Introductory example: performance measures

- **Router/Server utilization**
  - **Let B(t) be one if the server is busy at time t and zero otherwise.**

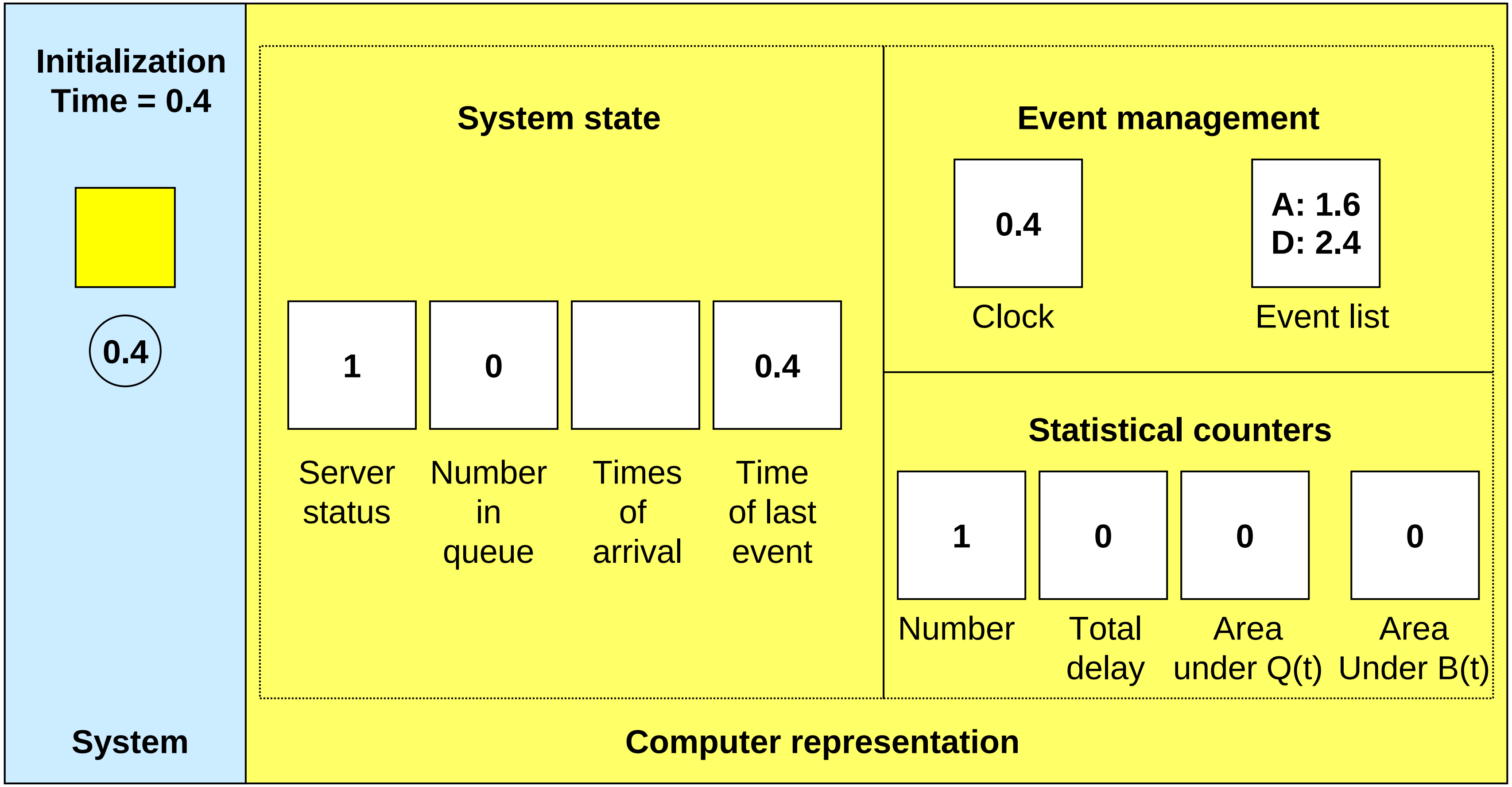


$$\hat{u}(n) = \frac{\int_0^{T(n)} B(t)dt}{T(n)}$$

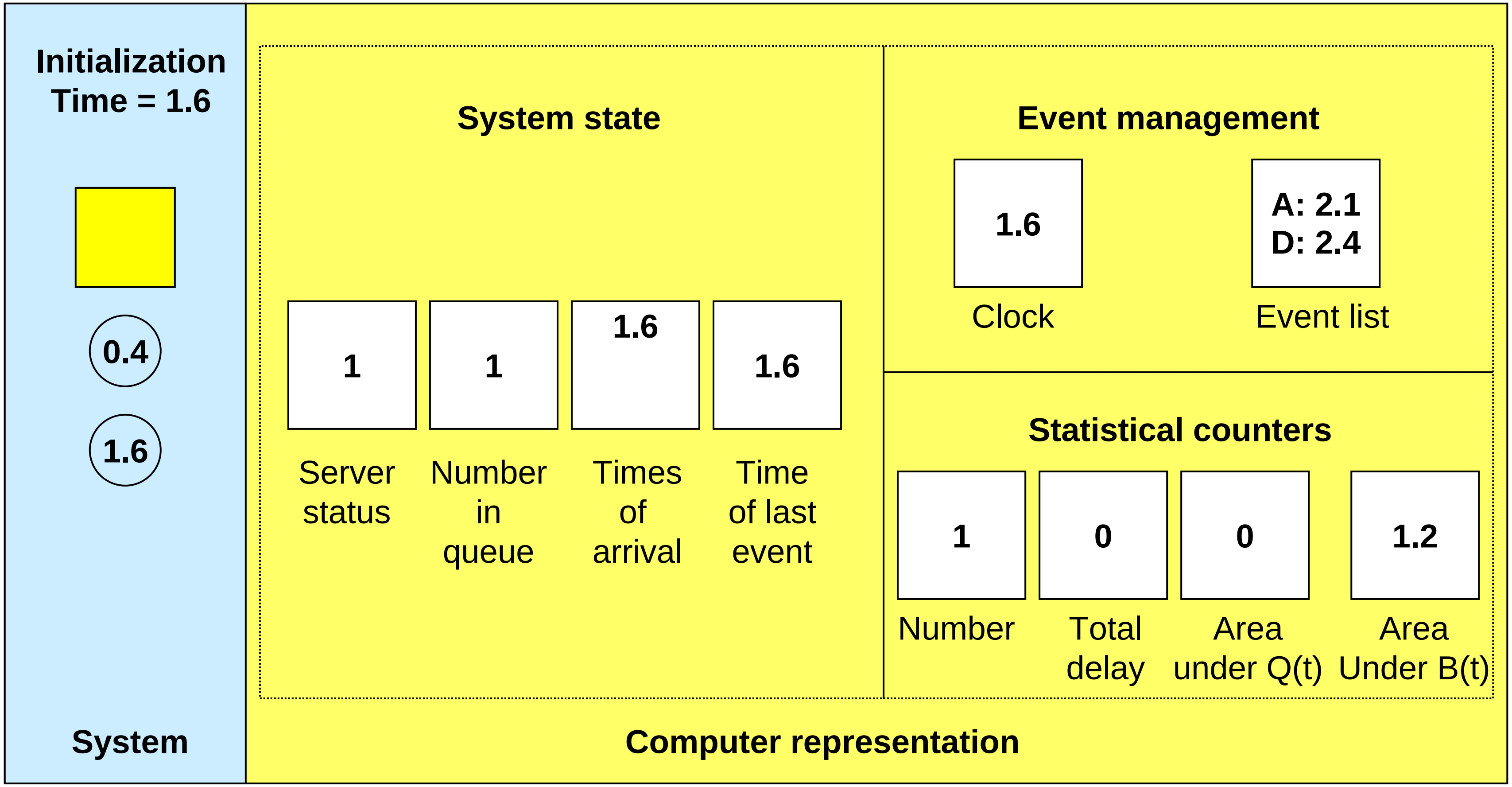
II Introductory example: execute model



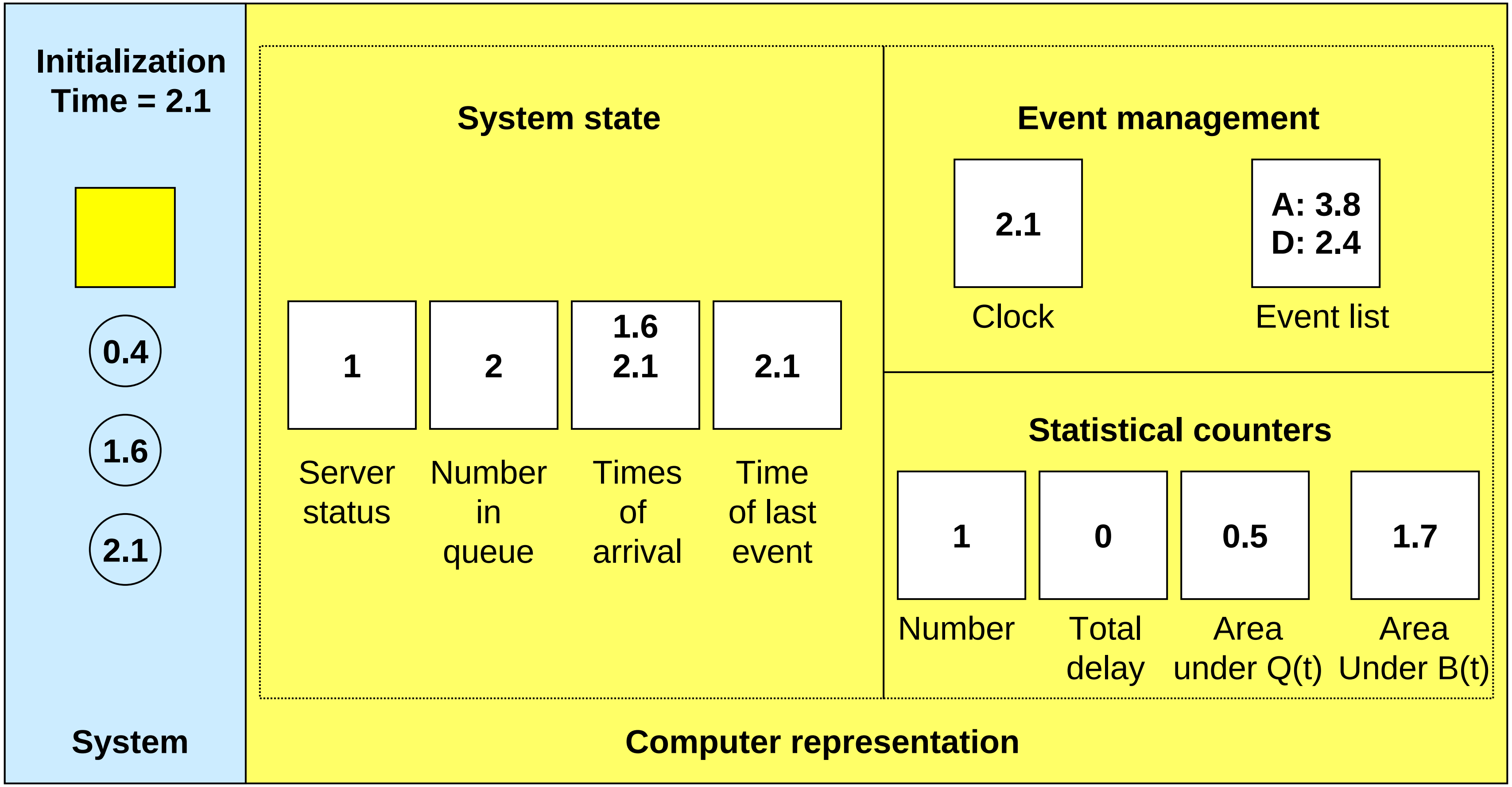
II Introductory example: execute model



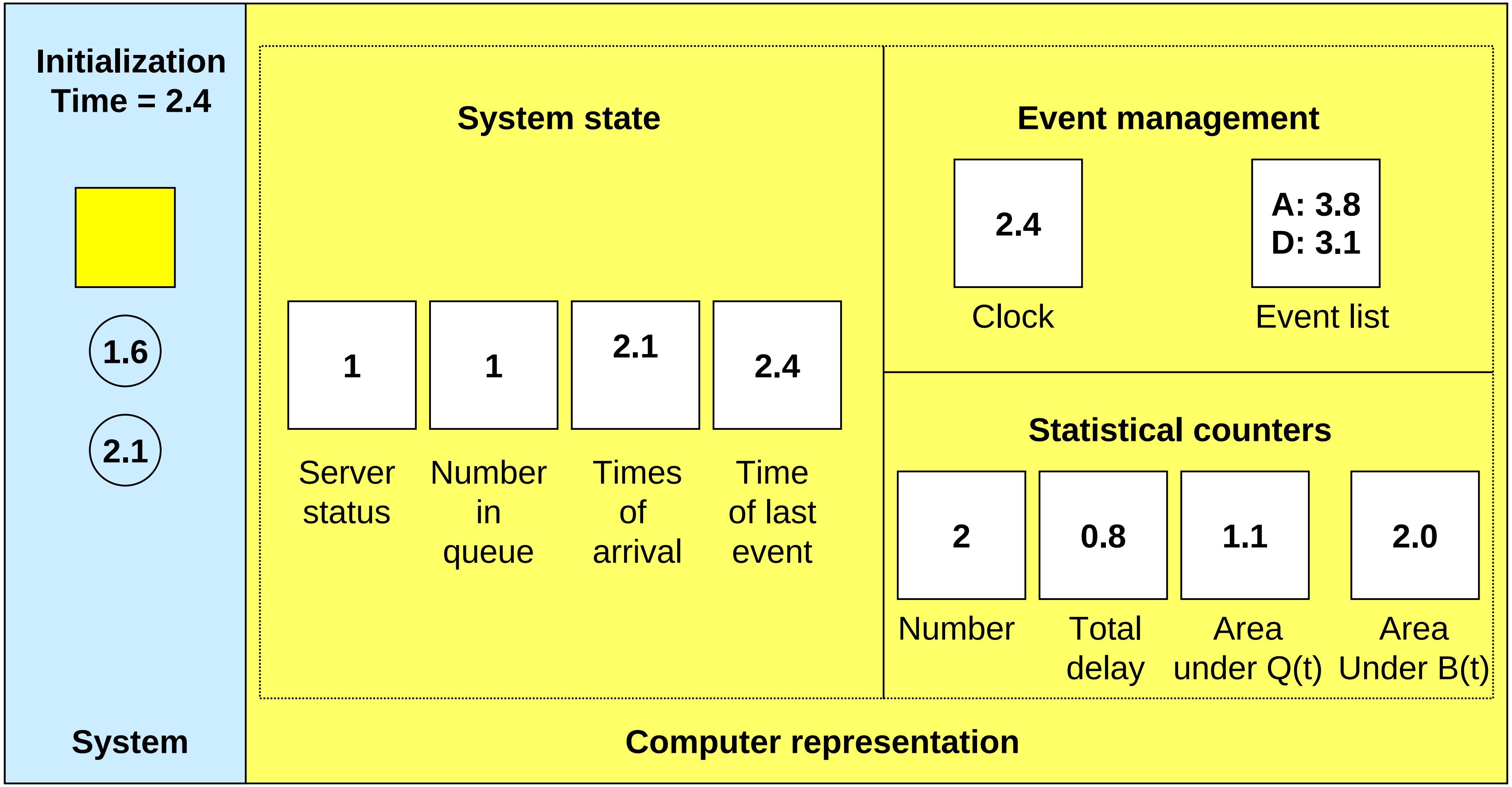
II Introductory example: execute model



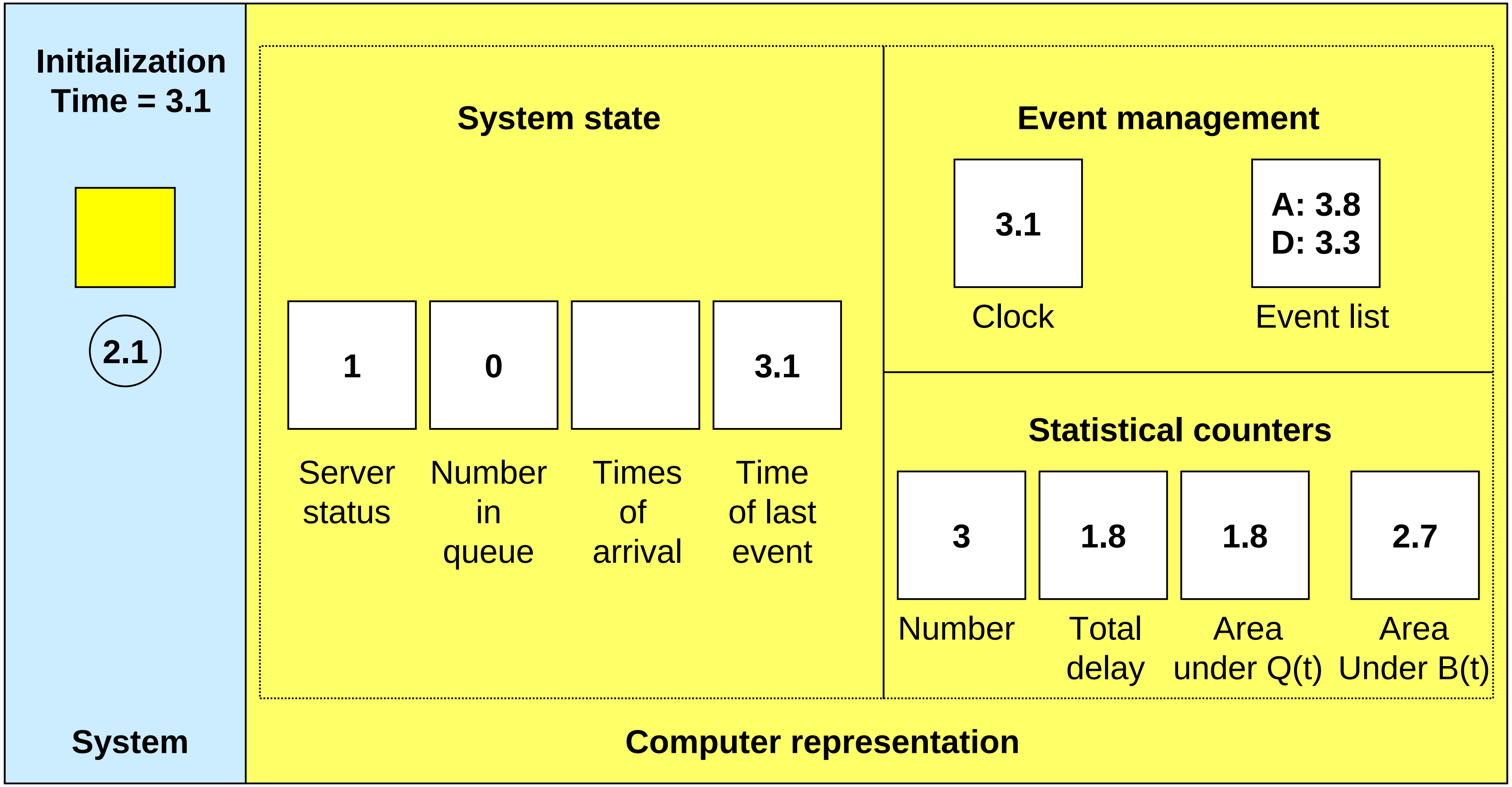
II Introductory example: execute model



II Introductory example: execute model

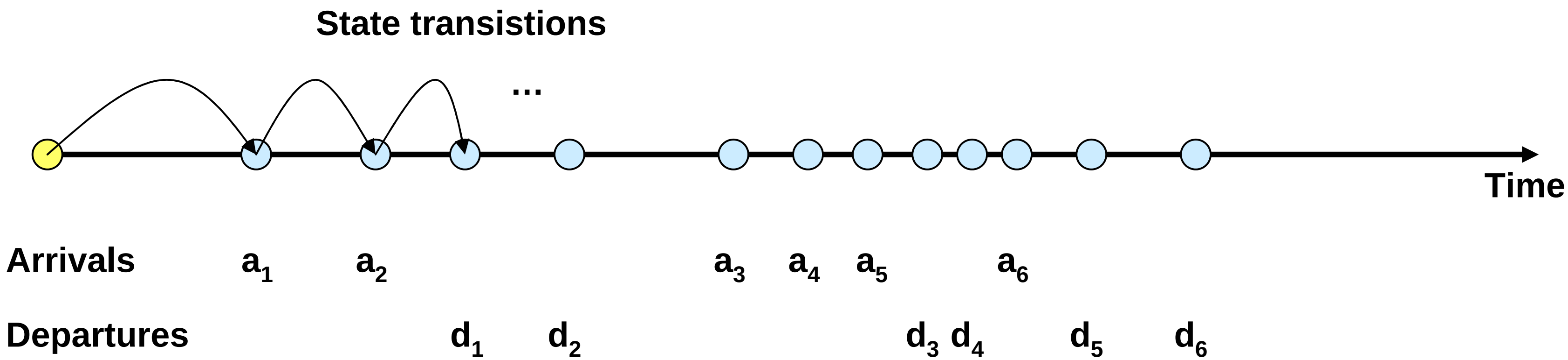


II Introductory example: execute model



… see [LK2000] for continuation of this example

II Introductory example: event logic



- Depending on the *event type,* a specific *event handler* is called that performs the appropriate state transition
- A state transition also includes generation of new events

II Introductory example: output

- n=1000
- Average delay in queue: 0.43 s
- Time-averaged number in queue: 0.418
- Server utilization: 0.46

  [Taken from LK2000]

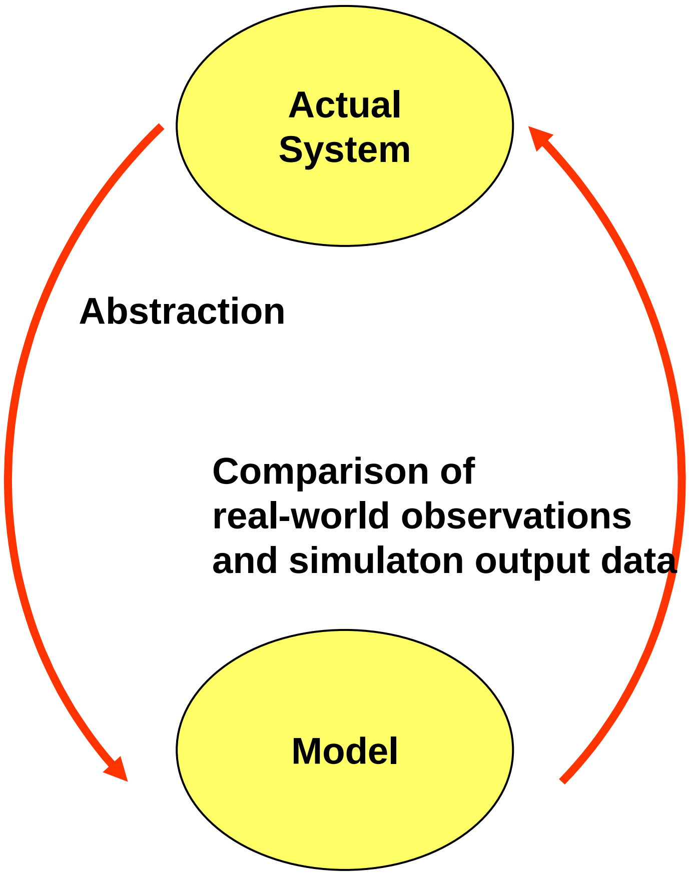
- Play with parameters in NetSim Lab 3!

III Computer networks: what is modeled and simulated?

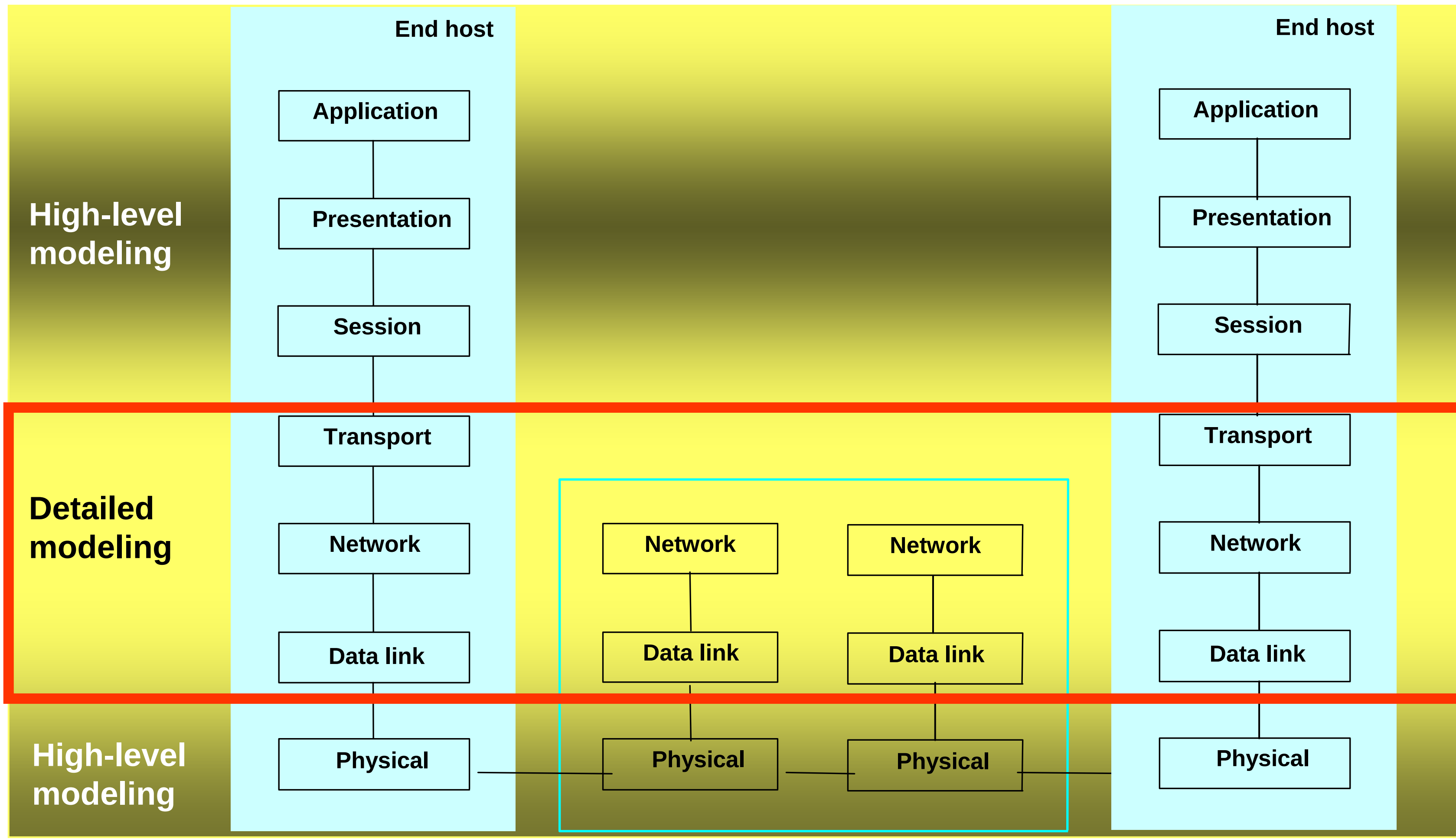
» **What is the ‘right’ level of detail?**

» **Usually, the following elements are modeled:**

- **Topology of the network, or mobility of the nodes**
- **Communication protocols as well as applications**
- **Data traffic**



III Simulation of computer networks: level of detail



One or more nodes within the network

III Trade off: accuracy vs computational costs

- **Packet-level simulation**
  - **No bit-level simulation as in digital communications**
- **Session-level simulation and aggregated flows**
  - **Coarser modeling than packet-level simulation**
  - **In movement pattern generation the difference goes by the name of microscopic vs macroscopic modeling (particles vs fluids)**
- **Internet-scale: for example, modelling of stub networks and autonomous systems (AS) as a ‚single node‘.**

- **Modeling depends on what has to be analyzed and how much computational costs one can afford.**

III Elements required for simulation of computer networks

» **Modeling**
  - Stochastic elements (generic, i.e., required for discrete event simulations of any system)
  - System states and logic (computer network-centric)
  - Statistics (generic, i.e., required for discrete event simulations of any system)

» **Simulation organization**
  - Event and time management, event handlers

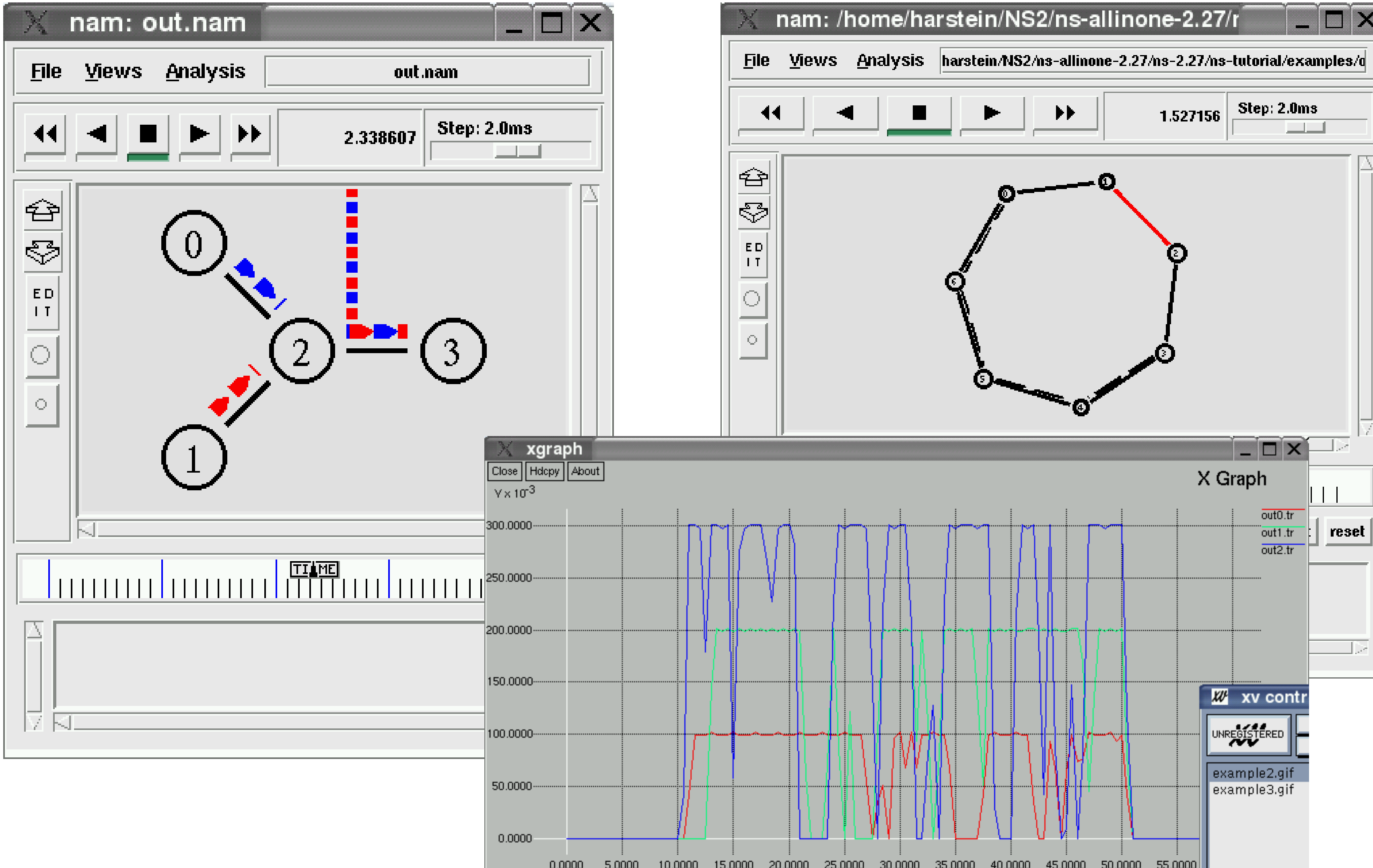
» **Output analysis**
  - Statistics

» **Tools**
  - For specifying scenarios
  - For running simulation
  - For analyzing simulation output

III Course overview

Modeling stochastic elements	**Random Number Generators, Generating discrete and continuous random variates Generating topologies, movement pattern, link characteristics, data traffic patterns**
Algorithmics of discrete event simulation	**Lists, heaps, calendar queues**
Modeling system state and logic for computer networks simulation	**Example: network simulator NS-2 nodes, links (point-to-point), agents, packets, connectors, classifiers, queues, packet scheduling, link delays, LAN MAC error modeling, transport protocols, application agent, …**
Output analysis	**Output data analysis Statistics, Tools, Visualization**

III Outlook



nam: out.nam
File Views Analysis
out.nam
2.338607
Step: 2.0ms
0
1
2
3
nam: /home/harstein/NS2/ns-allinone-2.27/r
harstein/NS2/ns-allinone-2.27/ns-2.27/ns-tutorial/examples/d
1.527156
Step: 2.0ms
reset
xgraph
Close
Hdcpy
About
X Graph
Y x 10^-3
out0.tr
out1.tr
out2.tr
300.0000
250.0000
200.0000
150.0000
100.0000
50.0000
0.0000
0.0000
5.0000
10.0000
15.0000
20.0000
25.0000
30.0000
35.0000
40.0000
45.0000
50.0000
55.0000
xv contr
UNREGISTERED
example2.gif
example3.gif

Summary / Educational Goal

- **Simulation: abstract view**
  - Systems, models, system state, state transitions
  - Classification of simulation types
    - Static vs dynamic
    - Continuous vs discrete
    - Deterministic vs stochastic
- **Simulation: concrete view**
  - Example: M/M/1 queue
  - Execution ‘by hand’
- **Elements needed for discrete event simulation**
  - State variables
  - Event management
  - Statistics/Counters
  - Generation of events
  - Transistion rules