IKR SimLib –
The IKR Simulation Library

An Overview

mail@ikr.uni-stuttgart.de
August 2008

Universität Stuttgart
Institute of Communication Networks
and Computer Engineering (IKR)
Prof. Dr.-Ing. Dr. h.c. mult. P. J. Kühn

Editors

Stefan Bodamer | Klaus Dolzer | Christoph Gauger
Marc Necker | Jörg Sommer | Joachim Scharf

Overview

- motivation
- classification of evaluation methods
- simulation basics
  - event-driven simulation
  - random number generation
  - statistical evaluation
- IKR simulation library
  - overview on the simulation environment
  - modelling concepts
  - measurement and evaluation
  - simulation control
  - parallel simulation
- practical usage
  - integration into development cycle
  - performing simulation studies
- extensions of the IKR simulation library
- sample problems solved using the IKR simulation library

Motivation

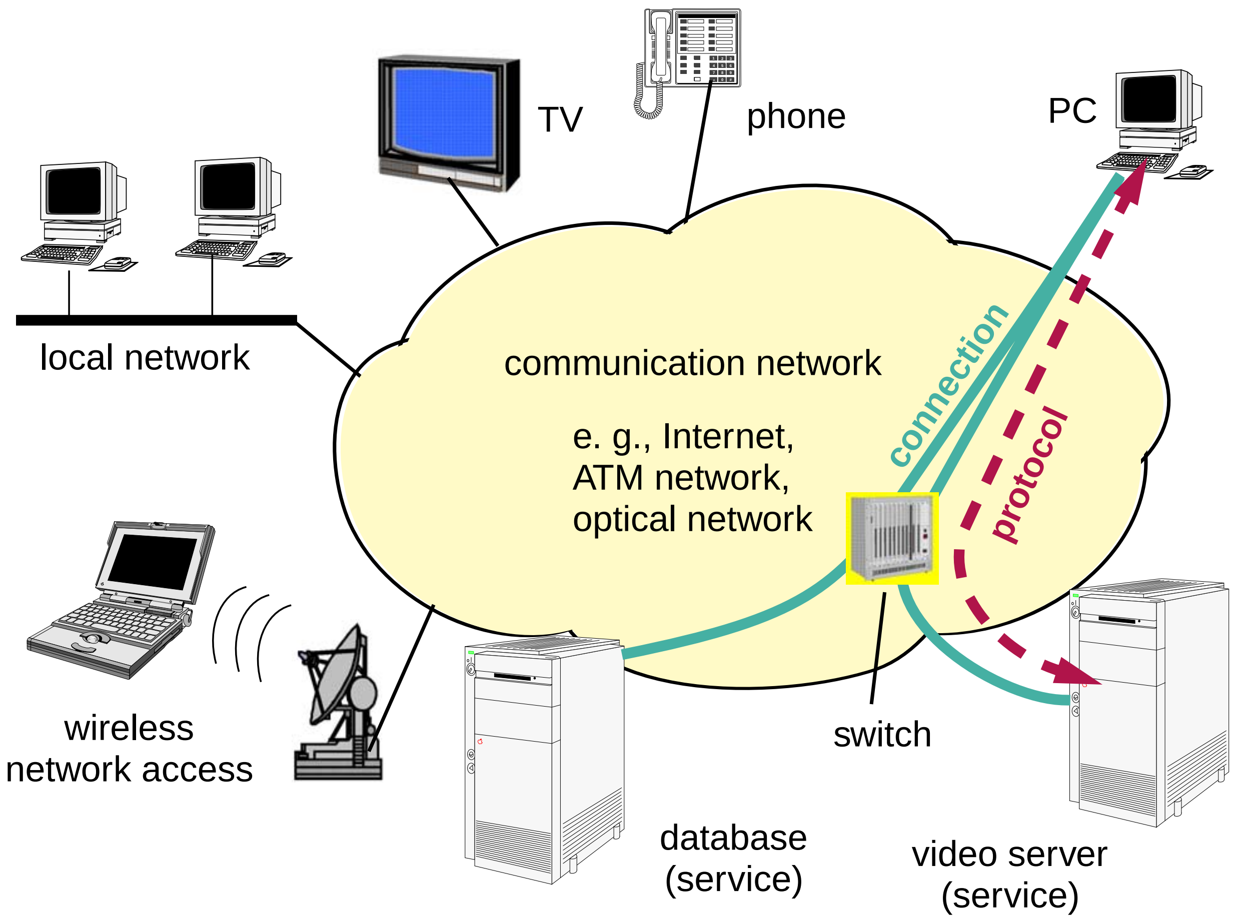
Communication Network Design Issues

requirements

- performance
  - response time
  - loss rate
  - throughput
  - fairness
  - signal quality
- availability
- security

influence on

- system dimensioning
- system structure
- protocols



Motivation

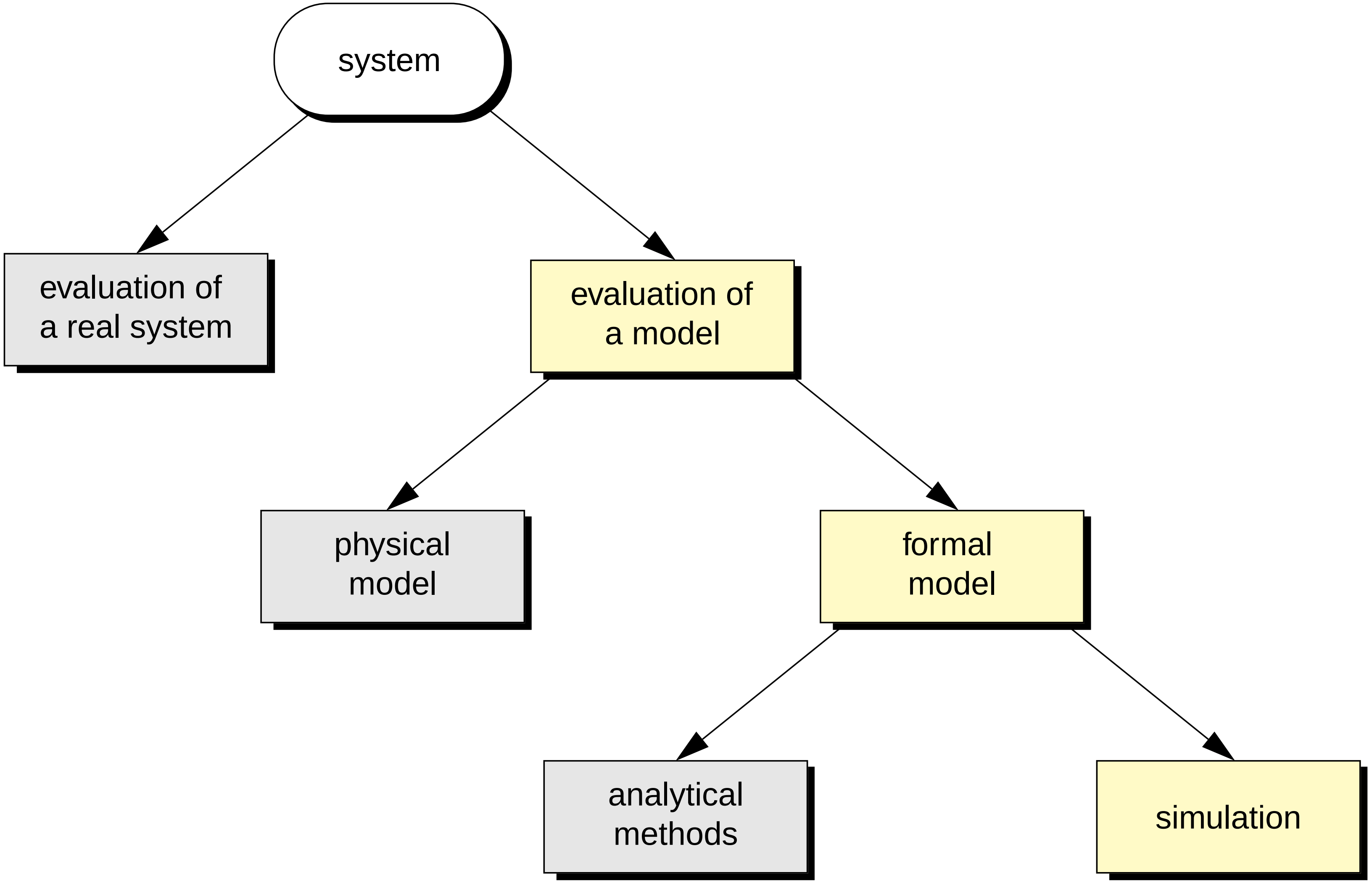
Simulation in Other Areas

examples

- computer systems evaluation
- analysis of production systems
- dimensioning and operation of transport systems (road network, airport, railway system, seaport)
- evaluation of operational organisation in hospitals, public offices, self-service restaurants, ...
- analysis of financial systems
- astronomy/meteorology
- fluid mechanics
- process sequence in a power plant
- aging process of system components

Classification of Evaluation Methods

Overview



according to [1.]

Classification of Evaluation Methods

Real System – Model

evaluation of a real system

- requirements
  - existence of the system
  - reproducibility
  - accessibility (space/time)
  - no danger
  - low cost
- examples
  - traffic measurements (telecommunication, road traffic)
  - evaluations with prototypes
  - tests of software products

evaluation of a model

- requirements
  - knowledge about the system (which parts are relevant?)
  - validity of the model
  - illustration
  - tractability
- characteristic measure

  grade of abstraction

Classification of Evaluation Methods

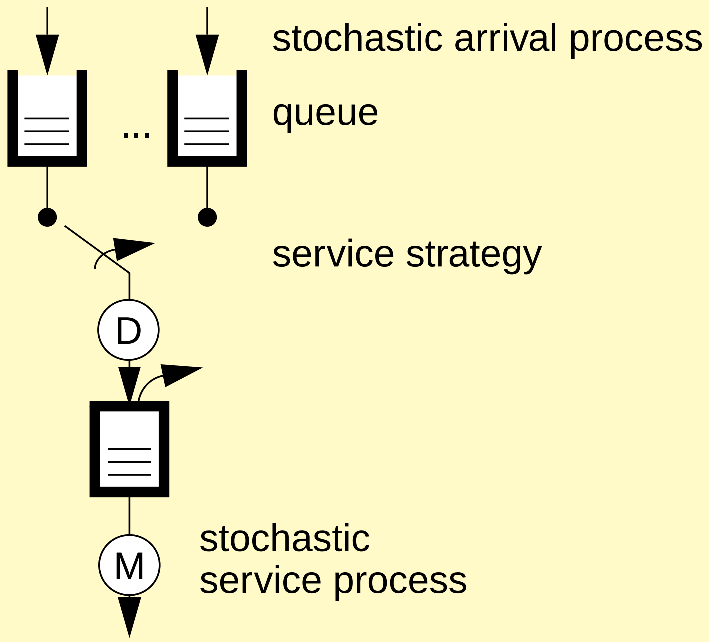
Physical Model – Formal Model

physical model

- physical imitation of the system
  - as a whole or partly
  - omitting irrelevant parts
  - simplification
- examples
  - automotive development (wind tunnel, crash test)
  - virtual cockpits
  - biological evaluations
  - sociology replay of situations by persons

formal model

- functional description
  - SDL
  - state transition diagrams
  - Petri nets
- queueing model



Classification of Evaluation Methods

Analytical Methods– Simulation

analytical methods

- exact or approximate solution of mathematical terms
- typical
  - systems of linear equations
  - differential equation systems
  - eigenvalue and boundary value problems
- critical
  - existence and robustness of a solution algorithm
  - complexity and accuracy of numerical evaluation

simulation

- replaying relevant procedures on a computer
- typical
  - high number of events
  - usage of random numbers
  - measurement of samples
  - statistical evaluation
- critical
  - statistical robustness
  - transient phases
  - memory/computation time
  - implementation faults

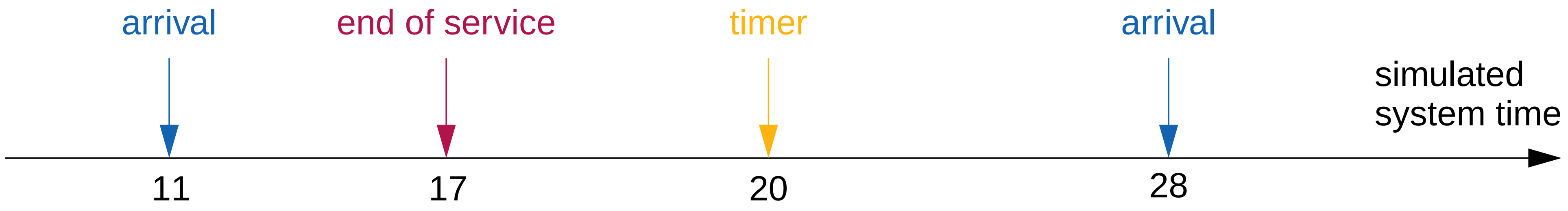
Classification of Evaluation Methods

Classification Criteria

- statistical – functional
- dynamic – static
- discrete – continuous
- event-driven – time-driven
- stochastic – deterministic
- stationary – instationary
- sequential – parallel
- in software – in hardware

Simulation Basics

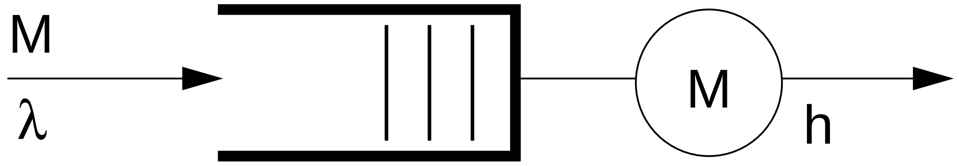
Event-Driven Simulation



- system state changes at discrete points in time (e.g. arrival of a message, end of service, timer expiration)
- state change represented by an event
- registration of events in a calender (together with a timestamp representing planned time of occurrence)
- sequential processing of events in the calendar by order of timestamps
  - update simulated system time
  - execute actions to realize state changes
  - plan successive events

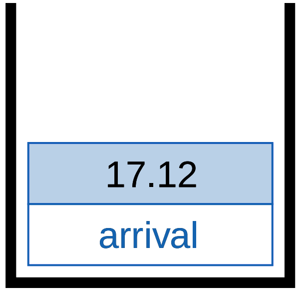
Simulation Basics

Example: M/M/1 Single Server System

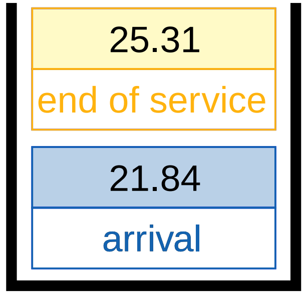


neg.-exp. distributed interarrival time with rate λ
neg.-exp. distributed service time with mean value h

- next event: arrival of a request
- new system time: 17.12
- state change
  - occupy server (if idle) or enter queue
  - update of statistics
- successive events
  - next arrival event + (possible) end of service
  - timestamp: current system time + random duration according to corresponding distribution
  - register in the calendar



calender



Simulation Basics

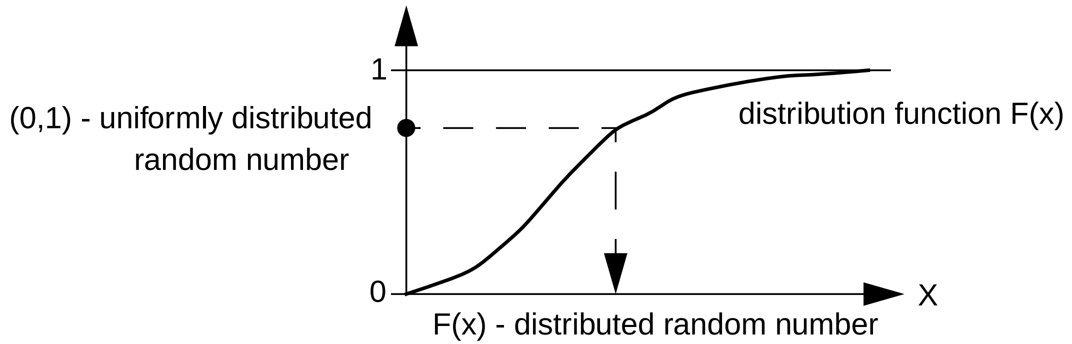
Generation of Random Numbers

- usage of random numbers, e.g. for
  - interarrival time
  - service time
  - batch size of batch arrivals
- classification according to nature of random numbers
  - continuous random variables (e.g. negative-exponential distribution)
  - discrete random variables(e.g. Poisson distribution, geometric distribution)
- generation of a (0,1)-uniform distributed random number
  - pseudo random number generator: deterministic algorithm for computing a random number with given starting value
  - example: linear congruential random number generator with sequence $Z_i = (a \cdot Z_{i-1} + c)$ mod m yields random no. $U_i = Z_i / m$ (e.g. $m = 2^{31} - 1$)
  - advantages: reproducibility, simple generation
  - disadvantages: periodicity, autocorrelation

Simulation Basics

Generation of Random Numbers (2)

- generation of a random number according to a distribution function
  - inversion of the distribution function $F(x) = P\{X \leq x\}$



  - example: random number for negative-exponential distribution function

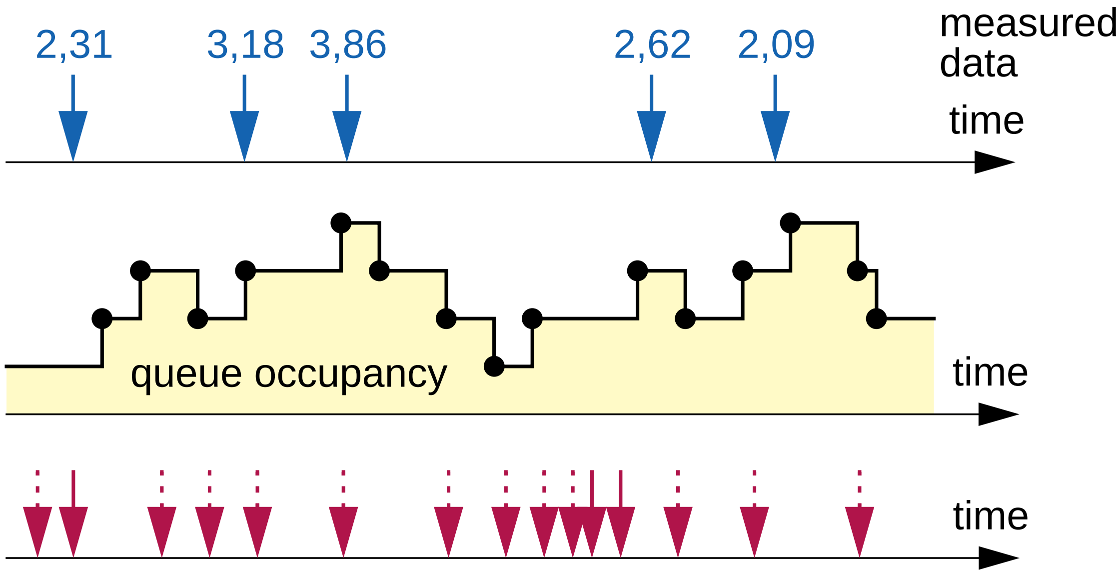
$$F(x) = 1 - e^{-\lambda x}: X = -\frac{1}{\lambda} \cdot \ln(1 - U) = -\frac{1}{\lambda} \cdot \ln U'$$

- alternatives
  - generator method: generation of a random number from several random numbers with invertible distribution (example: Erlang-k-distribution)
  - approximation (e.g. normal distribution)
- generation of values for correlated random variables according to a source model (e.g. On-Off, MMPP)

Simulation Basics

Statistical Evaluation

- examples for measured data
  - transfer time
  - queue length
  - loss



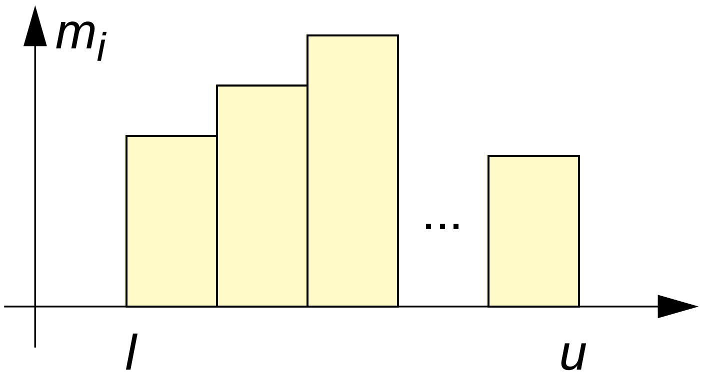
- evaluation according to
  - moments (mean value, variance, coefficient of variation)
  - relative occurrence (distribution and density)
  - quantile (e.g. 99% quantile $x_{0.99} : F(x_{0.99}) = P\{X \le x_{0.99}\} = 0.99$)
  - coefficient of (auto-)correlation/(auto-)covariance
- realization according to estimation functions or algorithms

Simulation Basics

Estimation of Statistical Measures

- mean: $\overline{X} = \frac{1}{m} \cdot \sum_{i=1}^{m} X_i$ (X_i: measured samples, m: number of samples)
- variance: $S^2 = \frac{1}{m-1} \cdot \sum_{i=1}^{m} (X_i - \overline{X})^2 = \frac{1}{m-1} \cdot \sum_{i=1}^{m} X_i^2 - \frac{1}{m \cdot (m-1)} \cdot \sum_{i=1}^{m} X_i$
- distribution (histogram)
  - divide relevant range of values $[l, u]$ into k intervals with size $w = \frac{(u-l)}{k}$
  - count number of samples falling into each interval: m_i $(i = 1, 2, \ldots, k)$
  - count number of samples smaller than l: m_l
  - estimator for $P\{X \leq l + j \cdot w\}$:

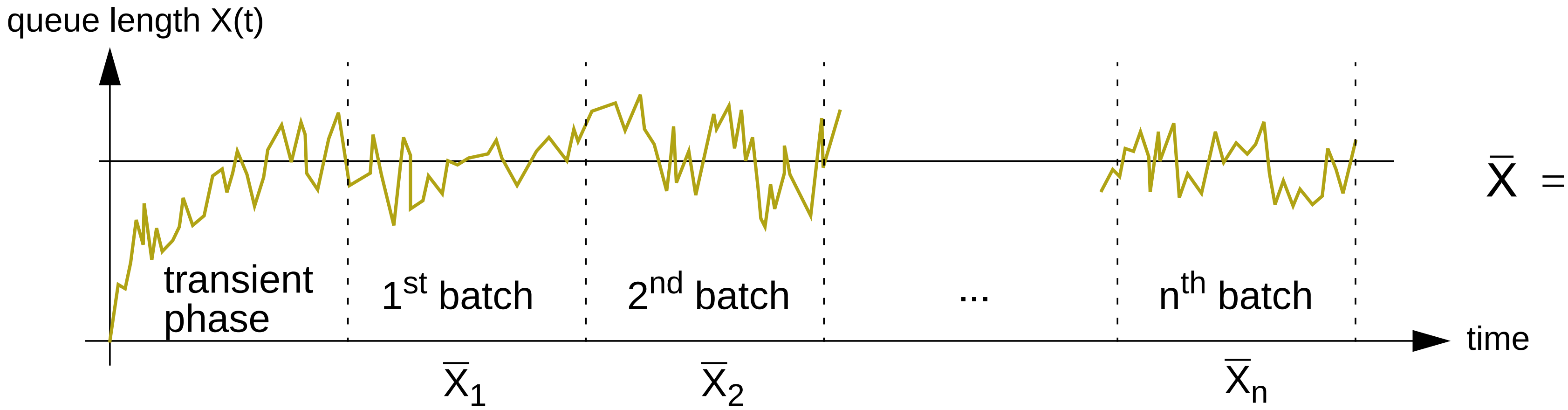
    $\frac{1}{m} \cdot \left(m_l + \sum_{i=1}^{j} m_i\right)$ for $j = 0, 1, \ldots, k$



Simulation Basics

Statistical Significance

- questions
  - how exact are the simulation results compared to the real values?
  - how long does a simulation have to last in order to reach sufficient exactness?
- commonly used: batch means method
  - partitioning of simulation into n batches (e.g. n = 10)
  - assumption of statistical independence between batches
  - interpretation of batch results as samples
  - averaging and computation of the confidence interval



$$\bar{X} = \frac{1}{n} \cdot \sum_{i=1}^{n} \bar{X}_i$$

Simulation Basics

Computation of Confidence Intervals

- related to confidence level α (e.g. α = 0.95: 95% confidence interval)
- interpretation: actual value μ is located within the confidence interval $[\bar{X} - \varepsilon, \bar{X} + \varepsilon]$ with probability α, $\rightarrow P\{\bar{X} - \varepsilon \le \mu \le \bar{X} + \varepsilon\} = \alpha$
- assumptions for batch results $\bar{X}_i$ (valid if batches are long):
- quasi-independent
- quasi-normal distribution (due to central limit theorem)
- random variable $T = (\bar{X} - \mu) / \sqrt{S^2 / n}$ is student-t-distributed with order $n - 1$

  (n = no. of batches, $S^2 = \frac{1}{n-1} \cdot \sum_{i=1}^{n} (\bar{X}_i - \bar{X})^2$ = empirical variance)
- ε and thus the confidence interval can be obtained from

$$\varepsilon = t_{n-1}\left(\frac{1+\alpha}{2}\right) \cdot \sqrt{\frac{S^2}{n}}$$

where $t_{n-1}(\beta)$ denotes β quantile of the student-t-distribution of order $n - 1$

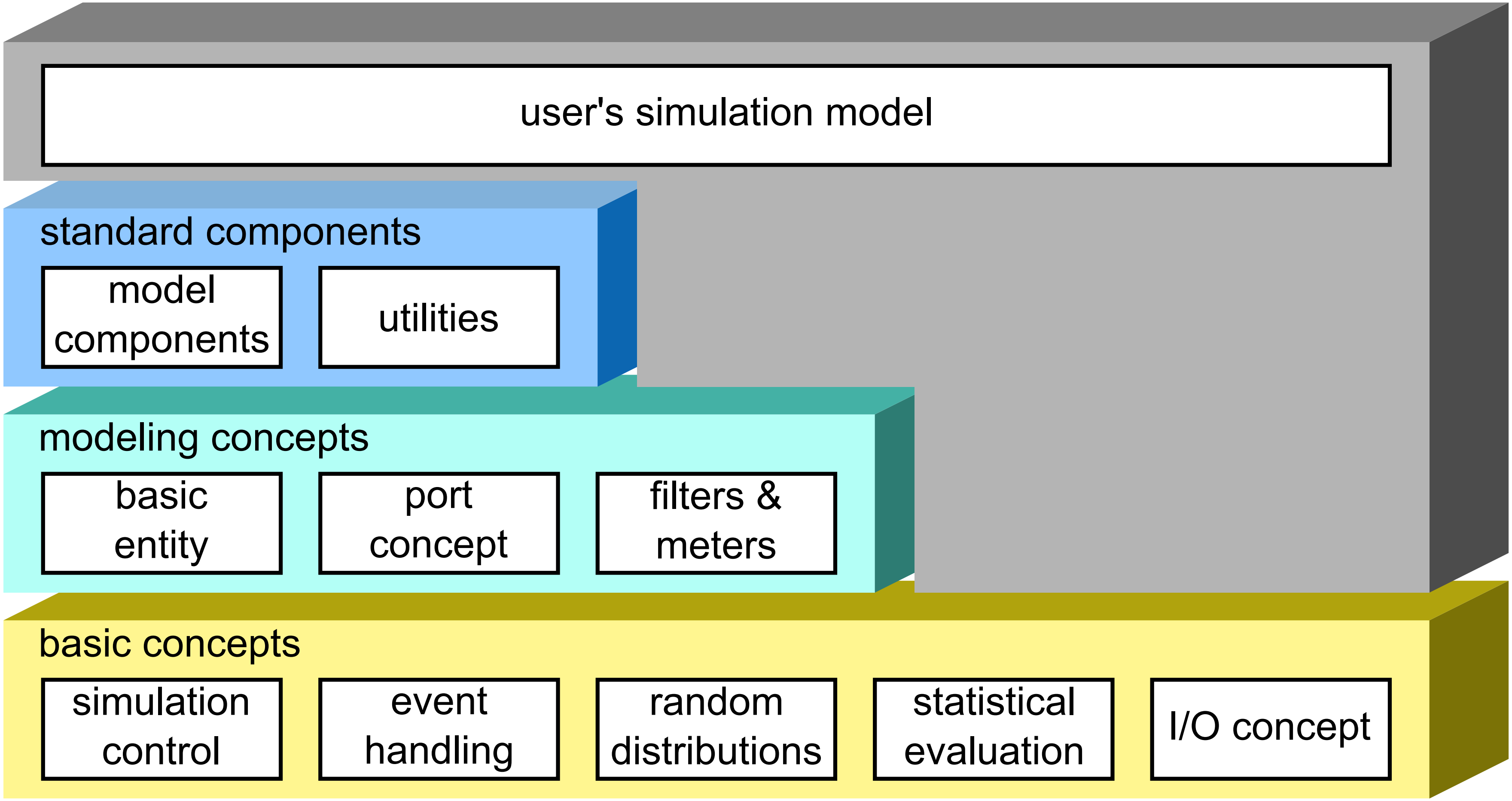
The IKR Simulation Library

Overview

- history
  - origin: Pascal simulation library (1980ies)
  - object-oriented redesign in the context of a dissertation (1992)
  - since then enhanced and improved continuously
- implementation
  - C++ class library (for class hierarchy see [2.])
  - Java library (for class hierarchy see [2.])
  - usage of additional libraries (e.g., container class library)
  - currently available for Linux, but almost no platform dependent code
- main features
  - support for transformation of an abstract model into source code
  - control of event-driven simulation
  - random number generation (various distributions and source models)
  - statistical evaluation
  - reading parameter values and printing results
- Responsible persons for support, maintenance, and development

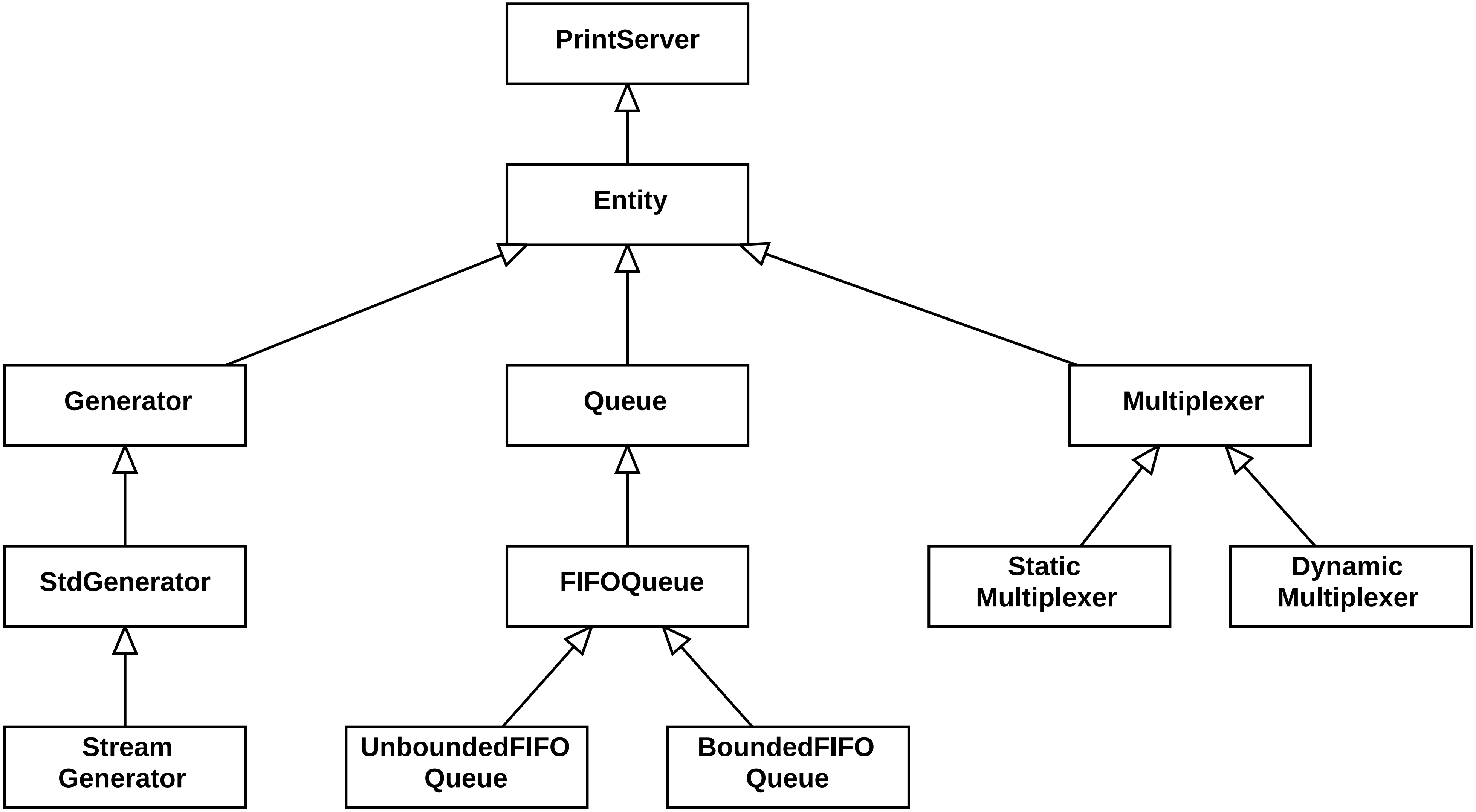
The IKR Simulation Library

Basic Structure



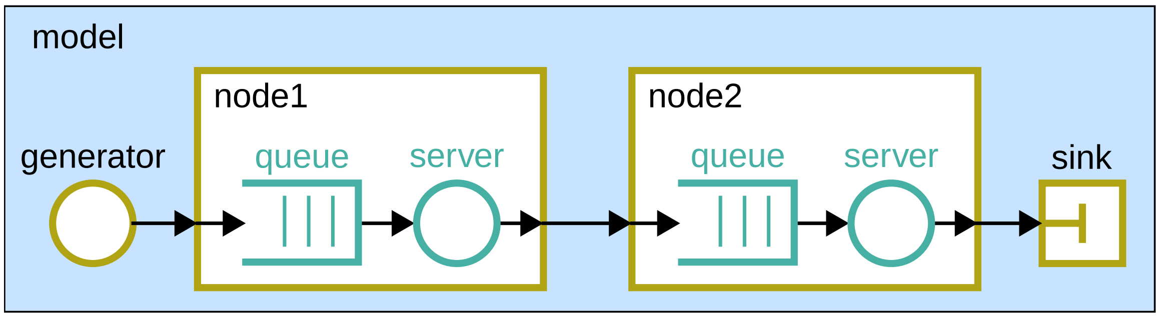
The IKR Simulation Library

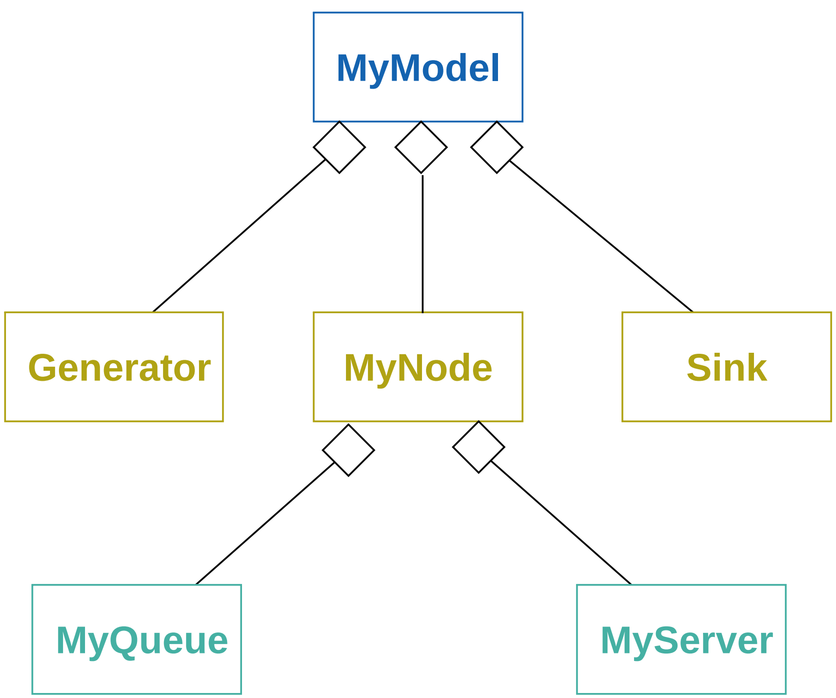
Model Components: Class Hierarchy



The IKR Simulation Library

Model Components: Object Hierarchy



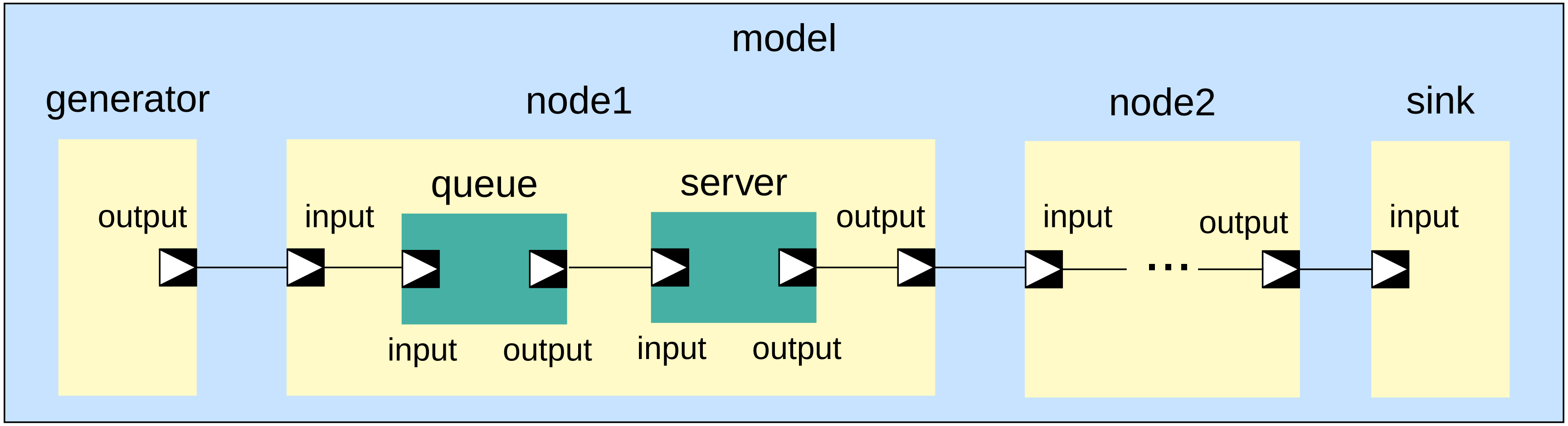


decomposition support by

- hierarchy
  - has-relationship
  - pointer to owner
- name concept
  - local name as attribute
  - identification of components
  - access via central component manager

The IKR Simulation Library

Modeling Components: Port Concept



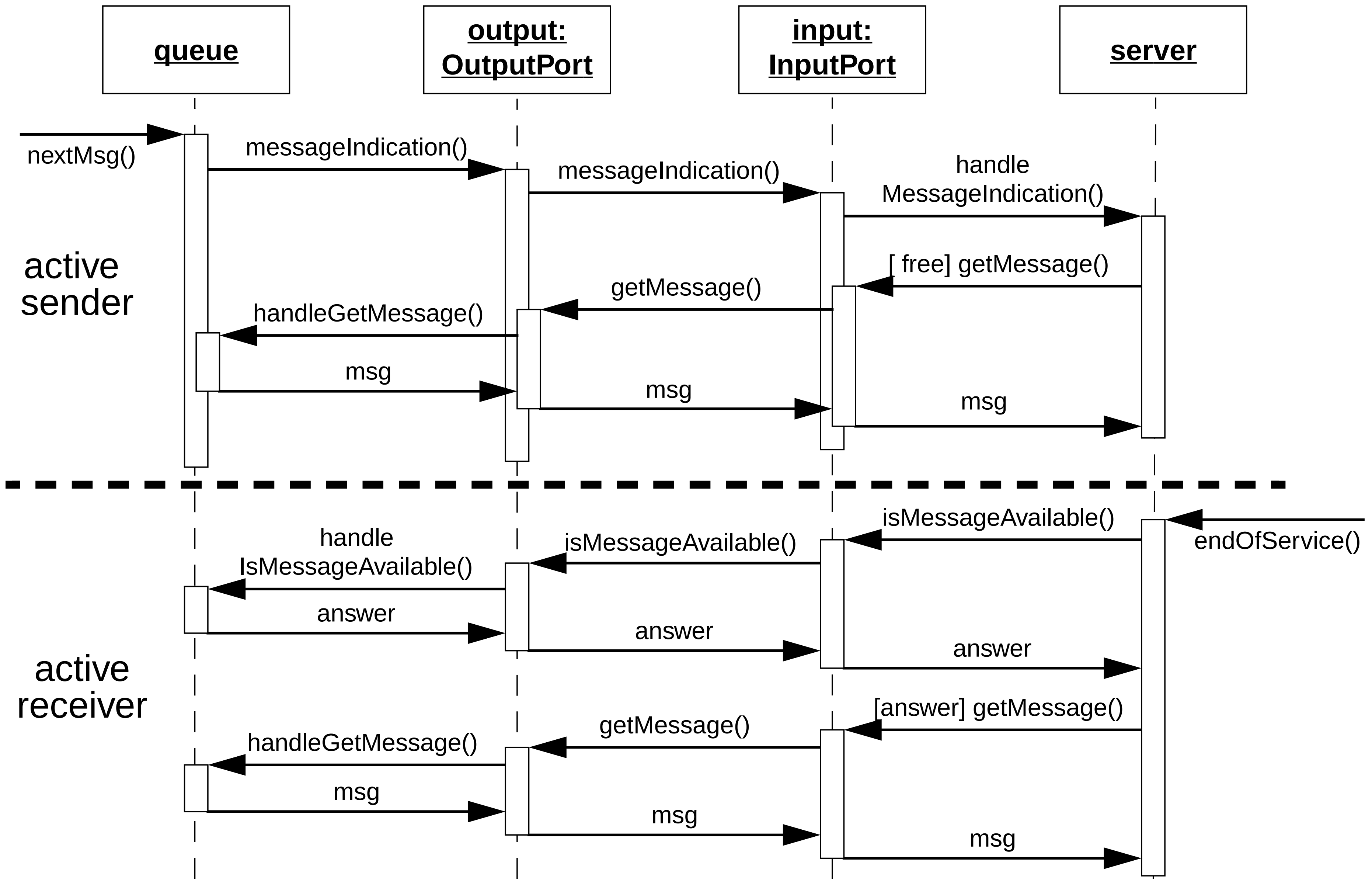
message exchange between model components via ports

- distinction between input and output ports
- central port registration using owner address and port name
- connection of the ports using function call `Connect`
- communication via handshake protocol

→ uniform interface for model components

The IKR Simulation Library

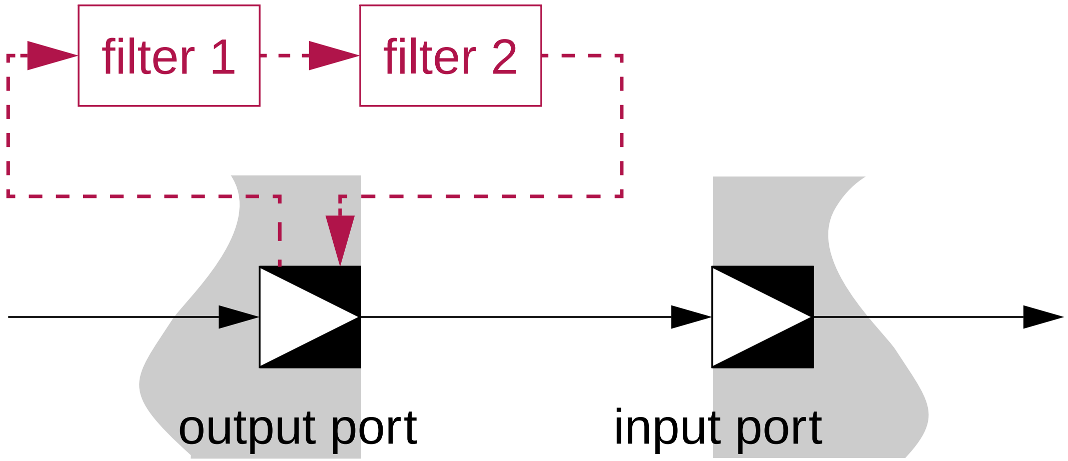
Exchanging Messages: Example



The IKR Simulation Library

Measurements: Message Filters

- attached to port using a method call
- redirection of protocol functions

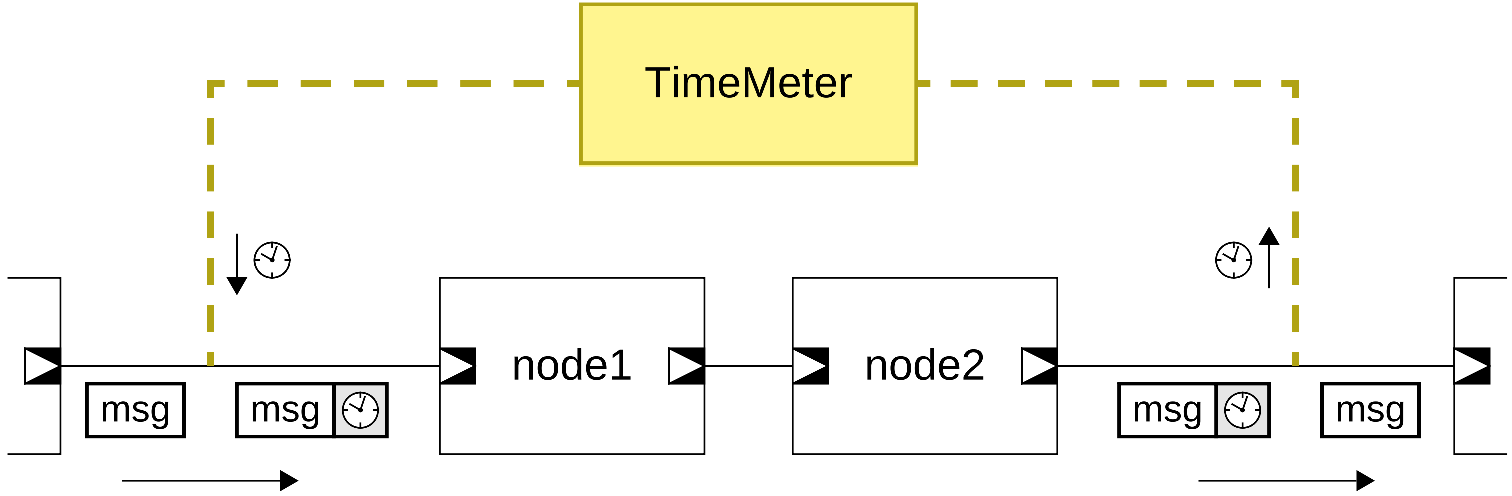


- motivation
  - evaluation or modification of message contents
  - message tracing
  - applied in meters

The IKR Simulation Library

Measurements: Meters

- attached to ports via message filters
- application
  - message counting/rate measurement
  - transfer time measurement

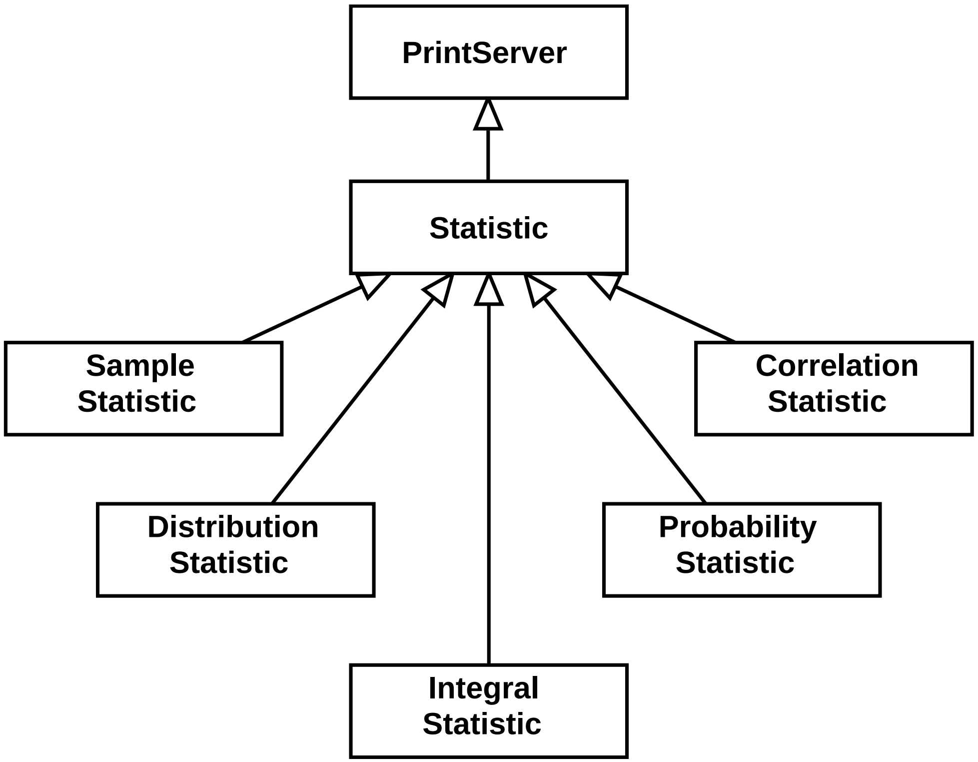


- internal evaluation of measured values using statistic classes

The IKR Simulation Library

Statistical Evaluation

- various statistic classes
  - sample moment (mean, variance)
  - time-based mean
  - quantile
  - probability
  - distribution (histogram)
  - correlation
- computation of confidence intervals
- application in
  - meters (e.g. transfer time)
  - model components (e.g. queue length)
- connection to simulation control



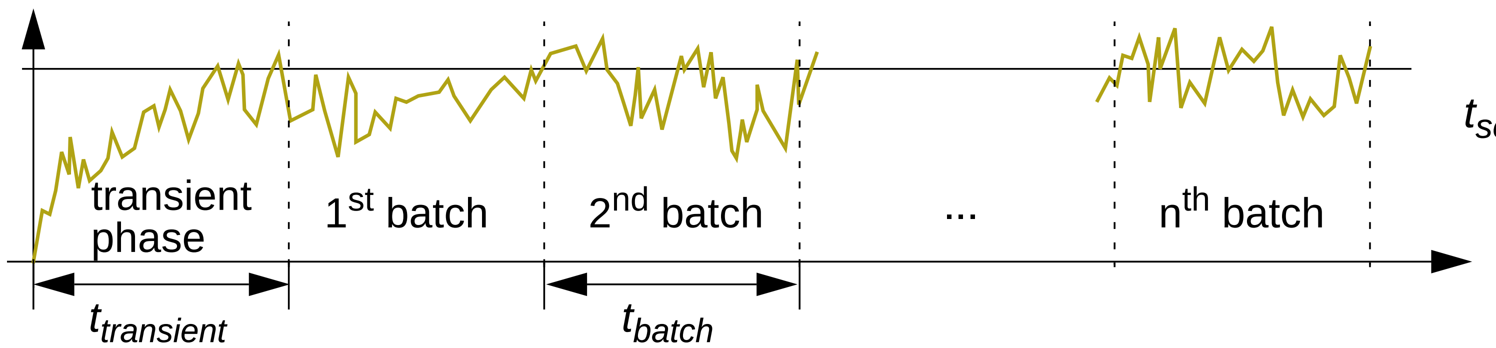
The IKR Simulation Library

Simulation Control

- execution of simulation by calling method `Run` in the simulation object
- initiation
- transient phase
- batches
- preparation and output of results
- access to simulation control via notifier
  - determine end of batch/simulation
  - based on message count or time
- triggering objects at beginning
  and end of phases, e.g.
  - statistics
  - generators

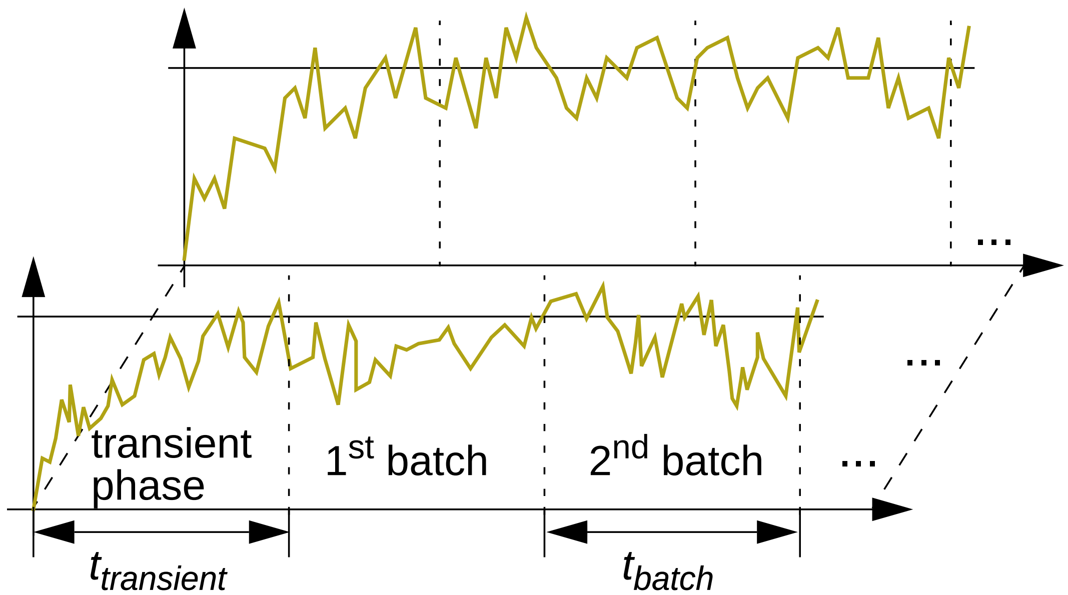
The IKR Simulation Library

Sequential Simulation



$t_{seq} = t_{transient} + n \cdot t_{batch}$

Parallel Simulation

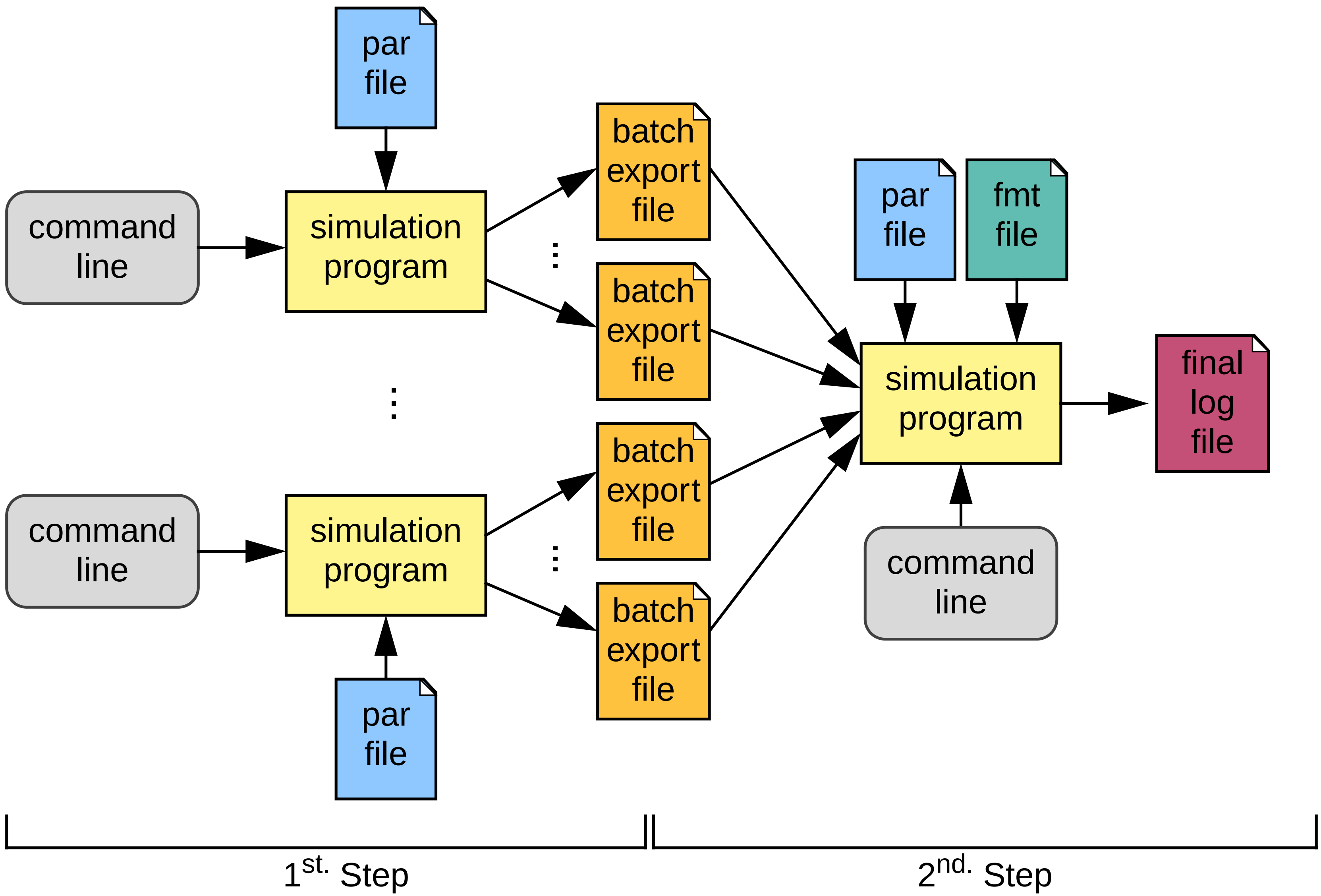


- parallel simulation in case of
  - simulation points with large simulation time
  - enough computing capacity
- independent simulation runs on different CPUs and/or cores

  → $t_{parallel} = t_{transient} + k \cdot t_{batch}$ with $1 \leq k \leq n$
- disadvantage: overhead due to multiple transient phases

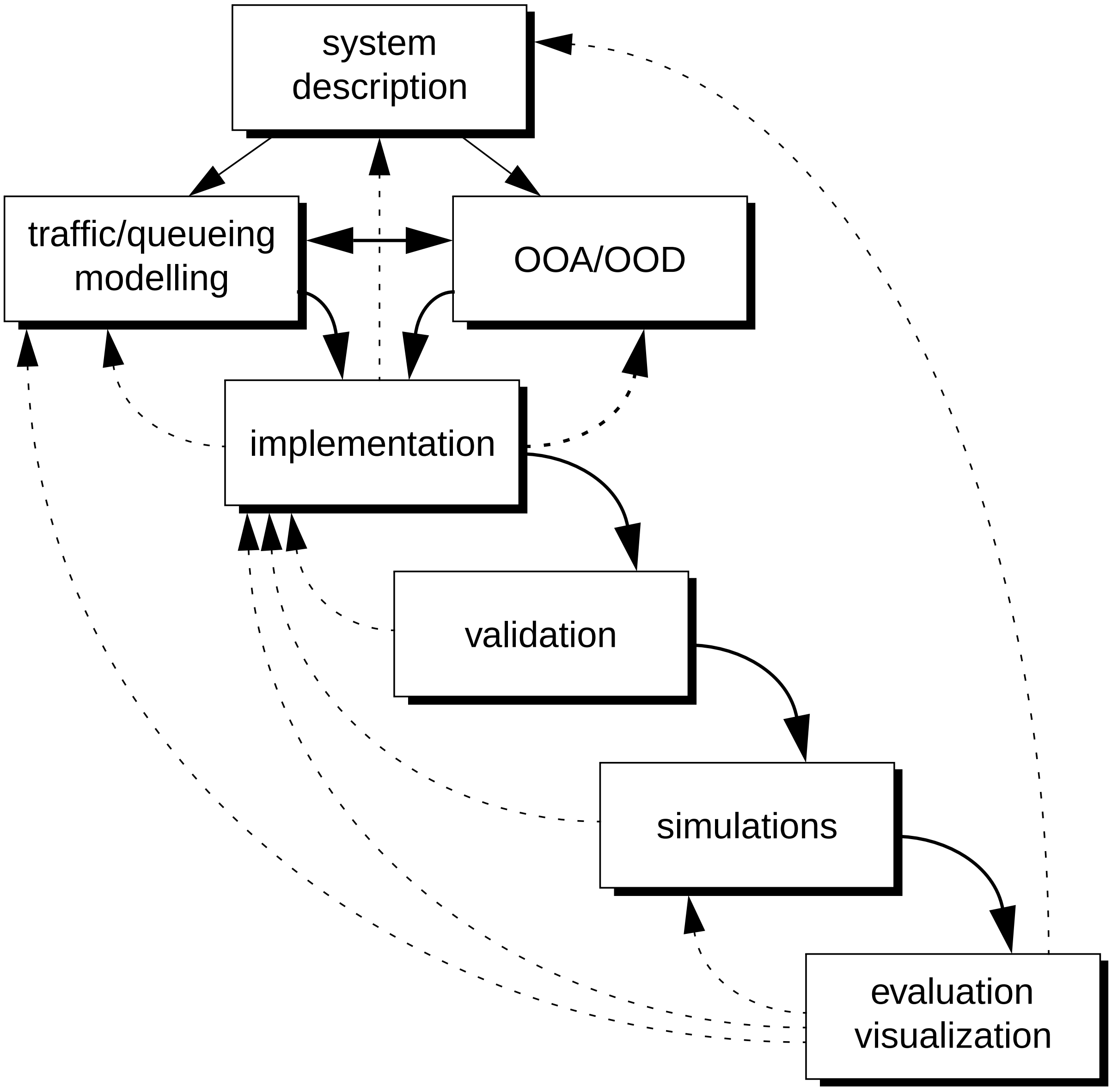
The IKR Simulation Library

Parallel Simulation: Realization



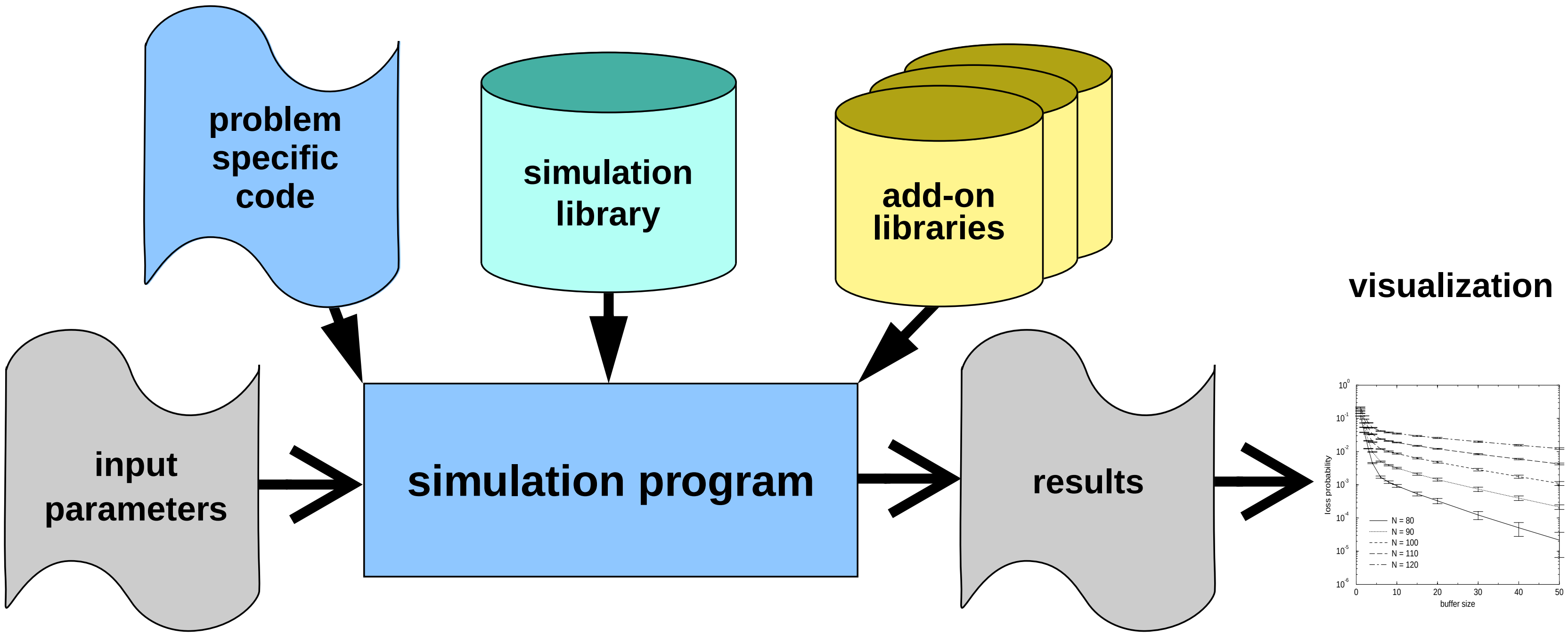
Practical Usage

Integration into Development Cycle



Practical Usage

Performing Simulation Studies

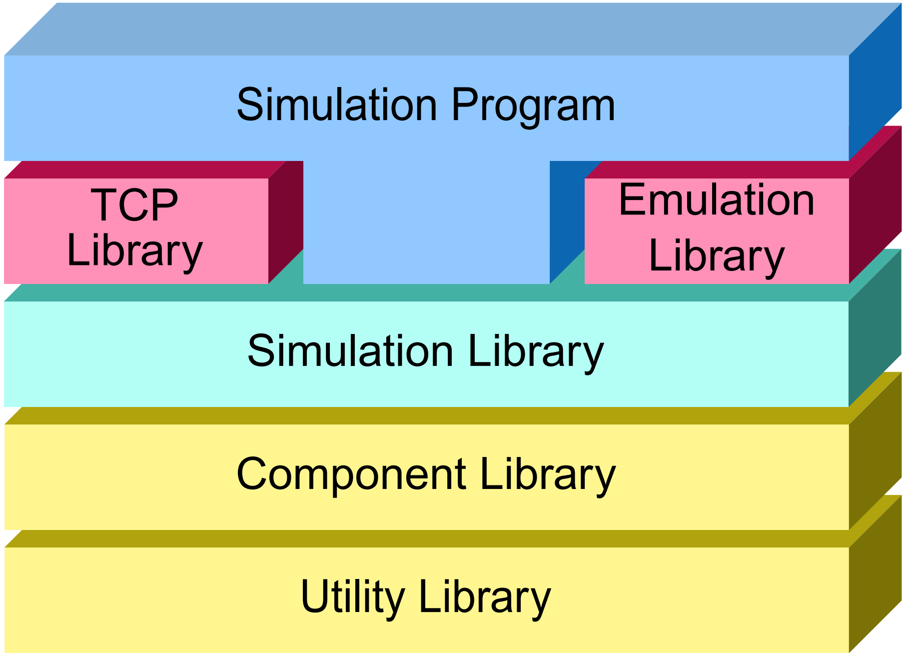


Practical Usage

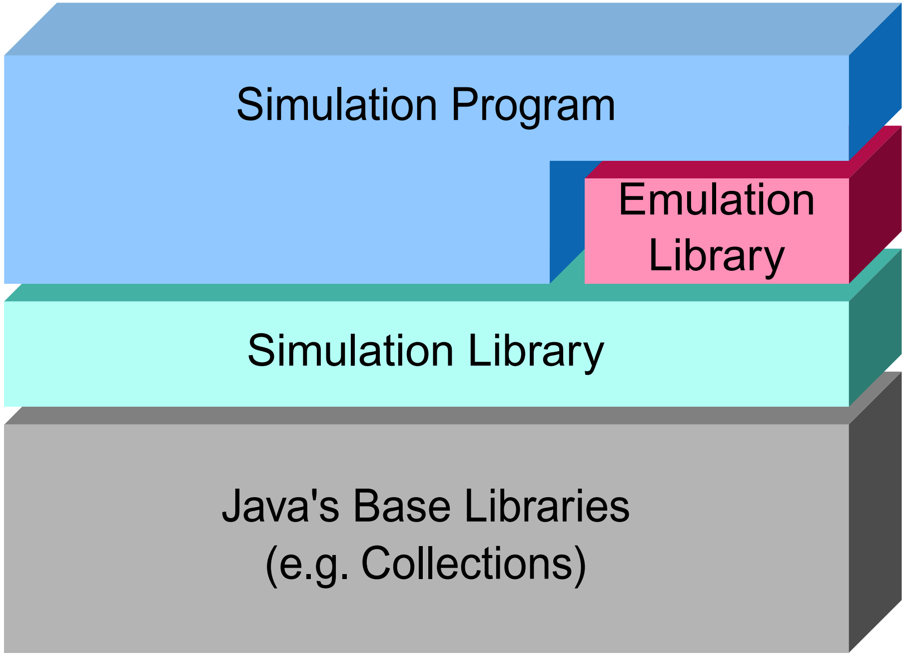
Simulation Programs

- problem-specific model and design
- typically several thousand lines of code

C++ Edition



Java Edition



libraries with more than 400 classes
and more than 70,000 lines of codes

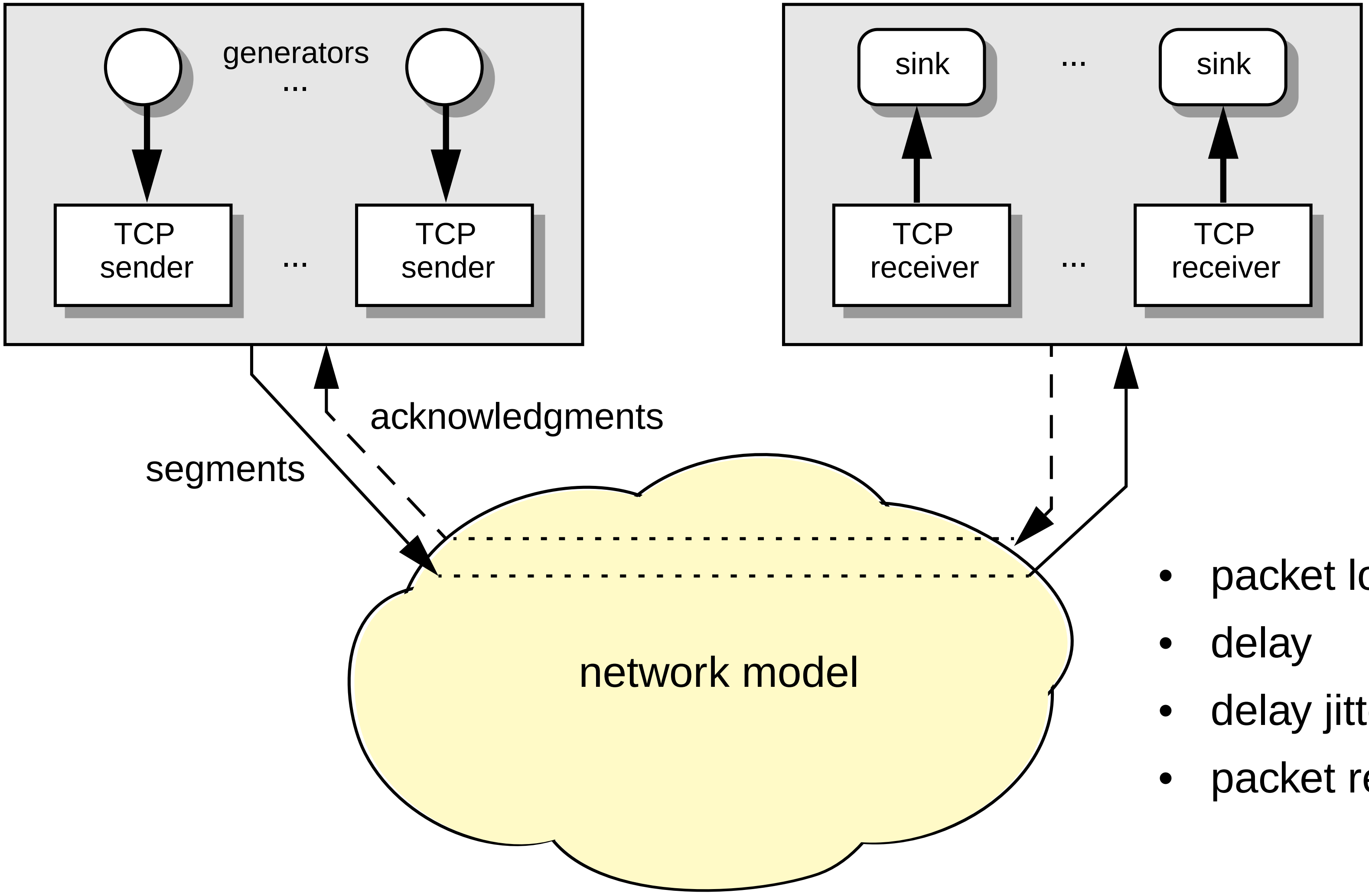
Extensions of the Simulation Library

TCP Library: Overview

- models unidirectional TCP data communication
- all important protocol mechanisms
  - reliable data delivery
  - flow control and congestion control
  - different TCP algorithms (Reno, NewReno, Tahoe, ...)
- light-weight implementation
  - simple connection setup and release
  - some enhancements are omitted, e. g., selective acknowledgments and explicit congestion notification
- flexible integration of application models
- results comparable to other simulation environments, such as ns-2 (UC Berkeley, LBL, USC/ISI, and Xerox PARC)
- currently, only available in C++ (and not in Java)

Extensions of the Simulation Library

TCP Library: Basic TCP Simulation Setup

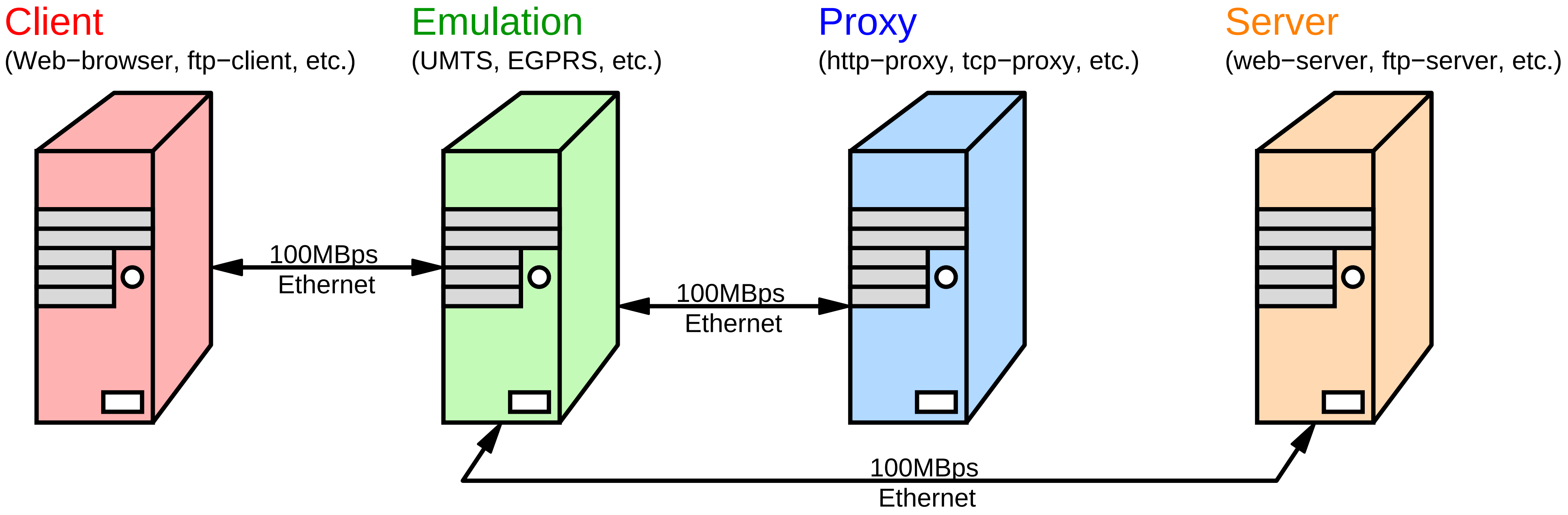


- packet loss
- delay
- delay jitter
- packet reordering

Extensions of the Simulation Library

Emulation Library

- integrated simulation and emulation environment
  - use simulation to optimize system over a large parameter space
  - use emulation to evaluate the performance of real world applications
- reuse of existing simulation tools
- example: WWW access over GPRS and UMTS



Simlib Usage in Projects

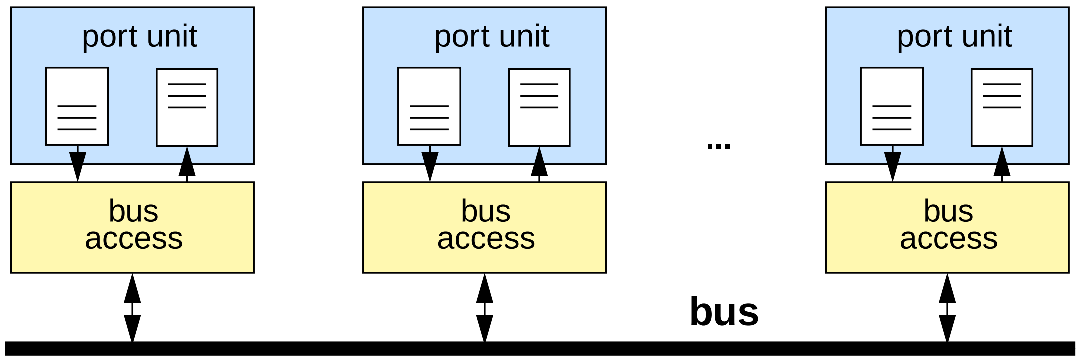
Overview

- optical networks
  - comparison of reservation concepts for optical burst switching
  - evaluation of dynamic routing strategies in WDM networks
- mobile communication networks
  - performance of packet traffic in radio access networks (GPRS, UMTS, ...)
  - capacity planning and network dimensioning
- Ethernet networks
  - dimensioning of switch buffers
  - evaluation of standard extensions (scheduling algorithms, frame extensions, ...)
- IP networks
  - IP networks with service differentiation
  - comparison of performance measures for TCP-based IP traffic
- ATM networks
  - dimensioning of ATM switches
  - investigation of traffic management mechanisms (CAC, ABR, GFR, ...)
- in-vehicle communication networks

Simlib Usage in Projects

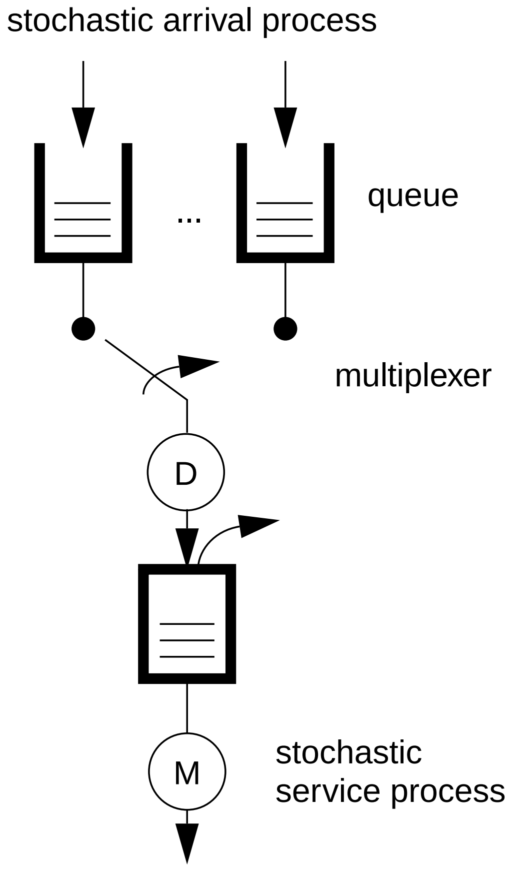
Example 1: ATM Switch

- system



- issues
  - cell losses
  - delay/jitter
  - influence of the bus protocol
  - buffer dimensioning

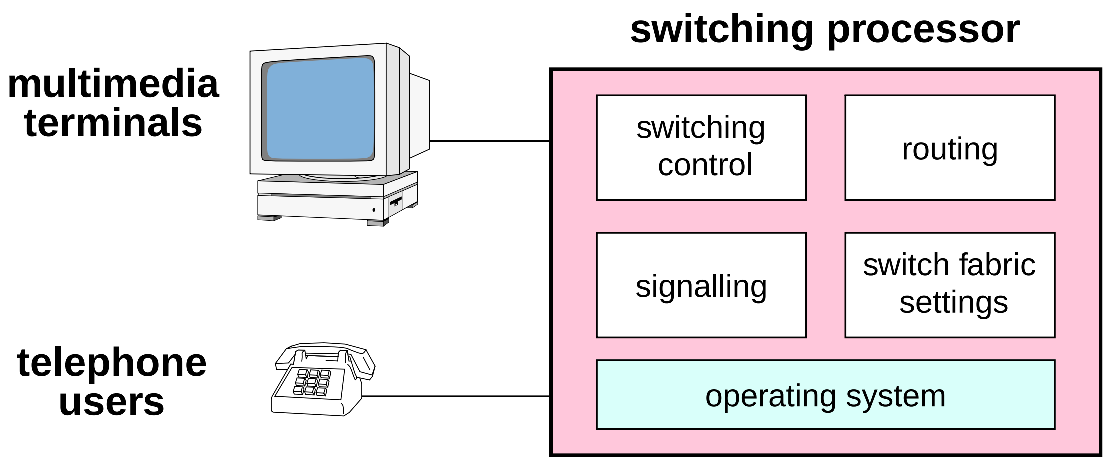
- modelling:
  queueing network



Simlib Usage in Projects

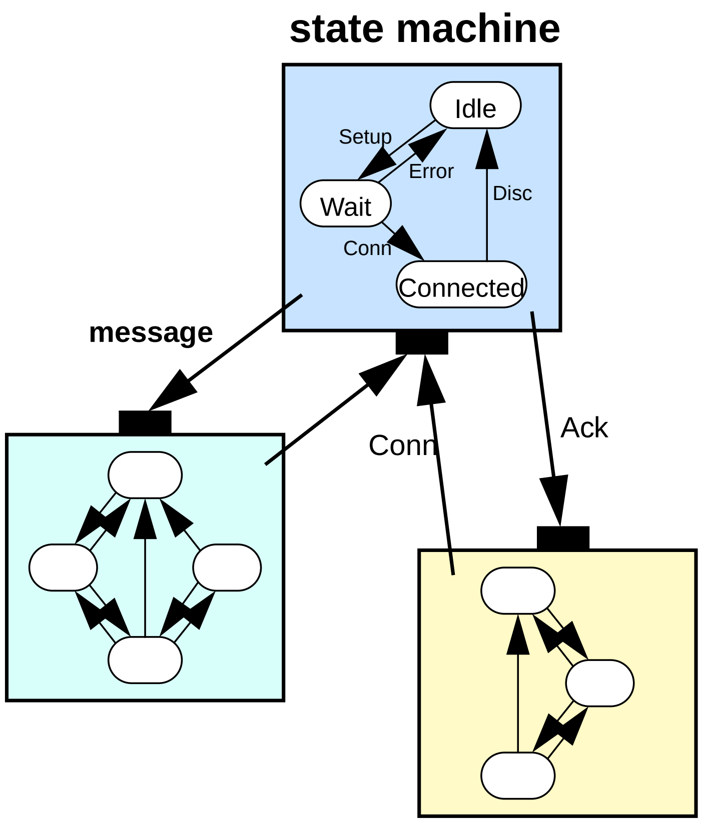
Example 2: Switching Software

- system



- issues
  - connection setup time
  - load caused by software components
  - overhead of operating system
  - influence of component distribution

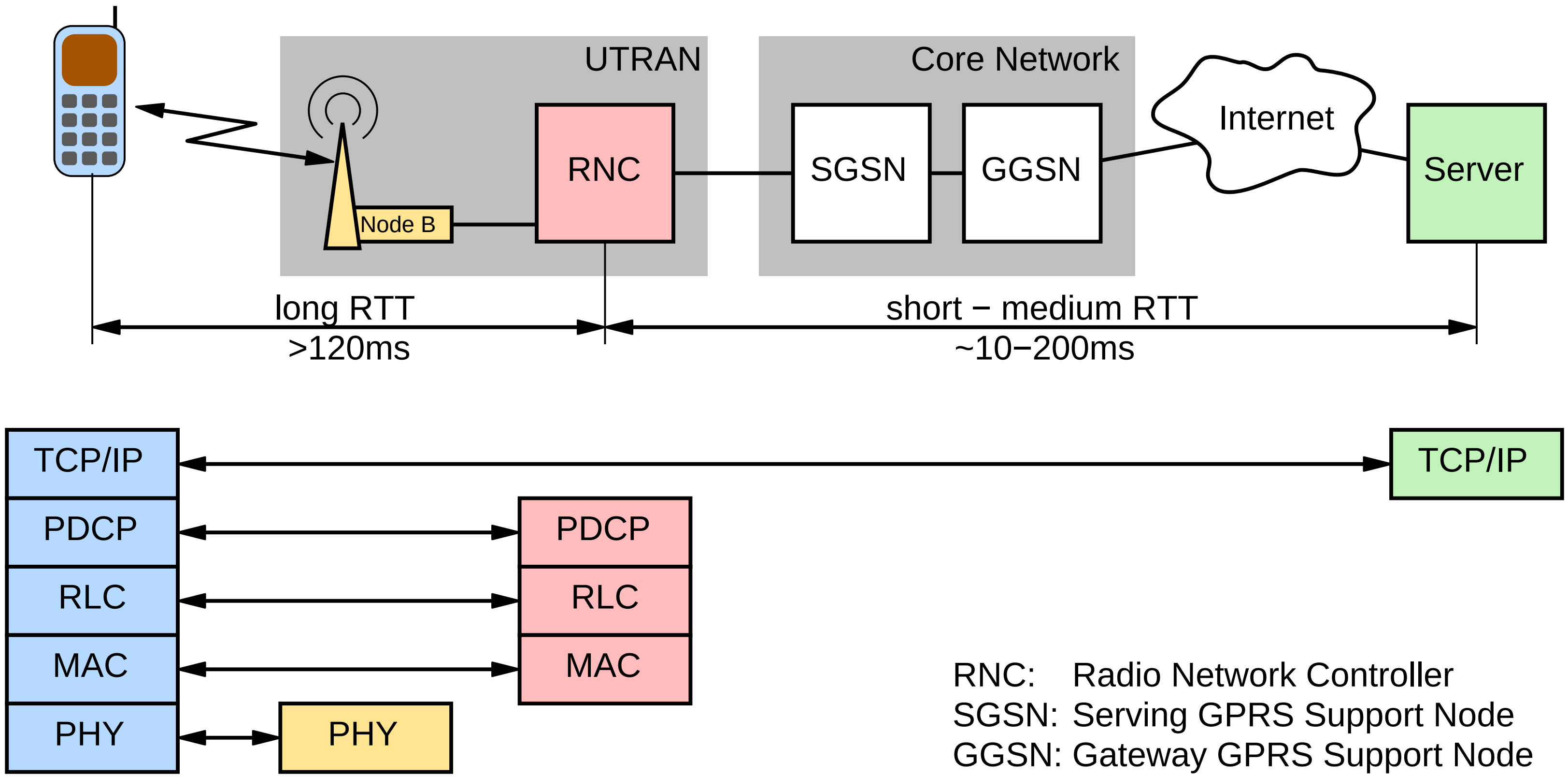
- modelling: state machine



Simlib Usage in Projects

Example 3: UMTS Performance

- system
- modelling: multi-layer model

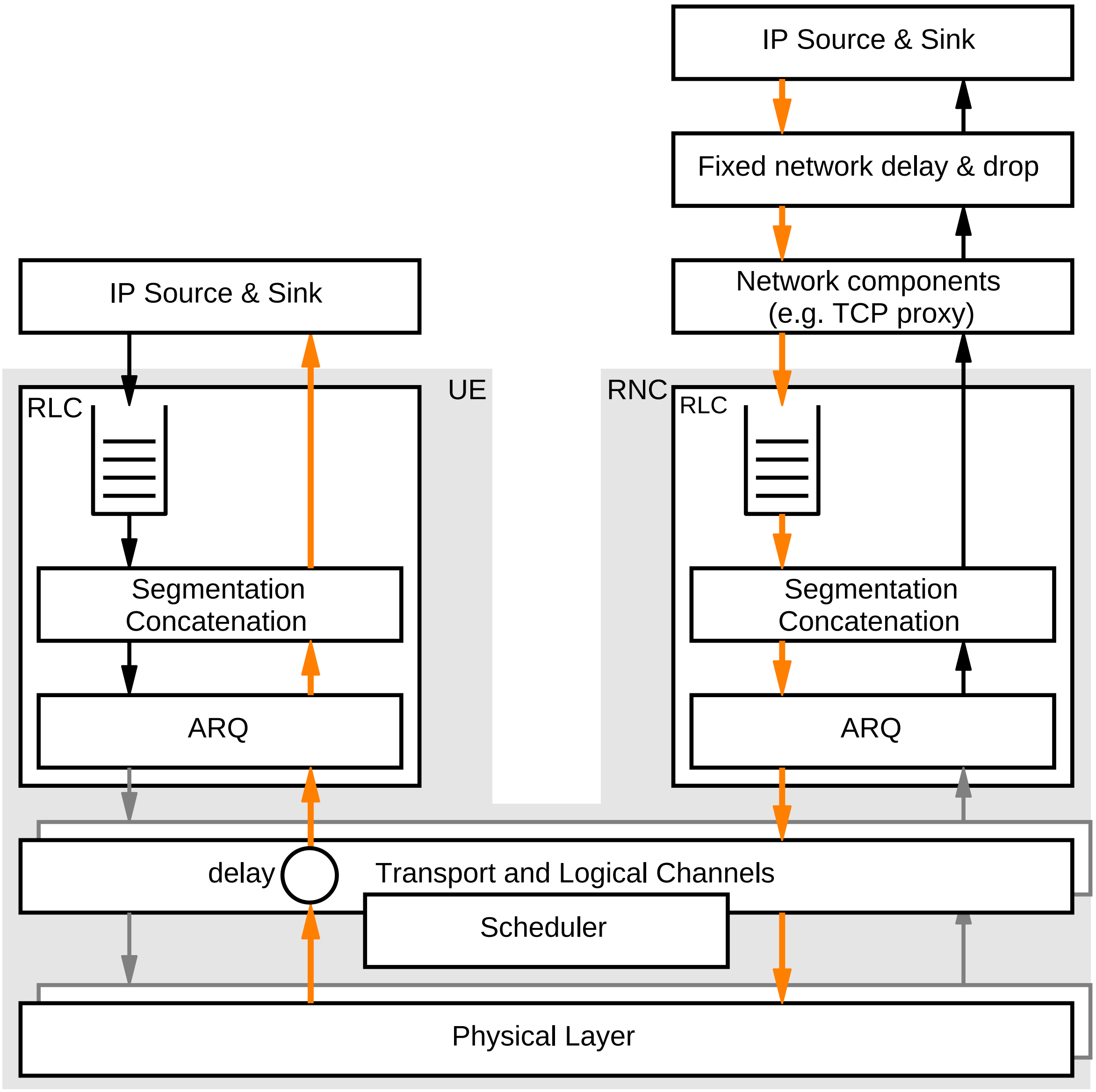


RNC: Radio Network Controller
SGSN: Serving GPRS Support Node
GGSN: Gateway GPRS Support Node

- issues
  - system dimensioning, cross-layer protocol optimization
  - transport and application layer performance

Simlib Usage in Projects

Example 3: Simulation/Emulation Model

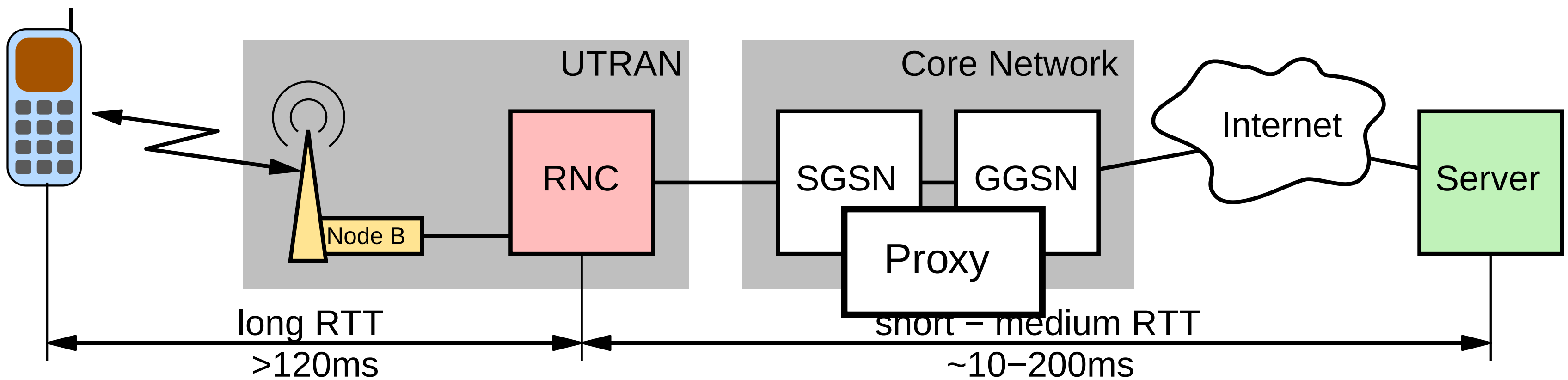


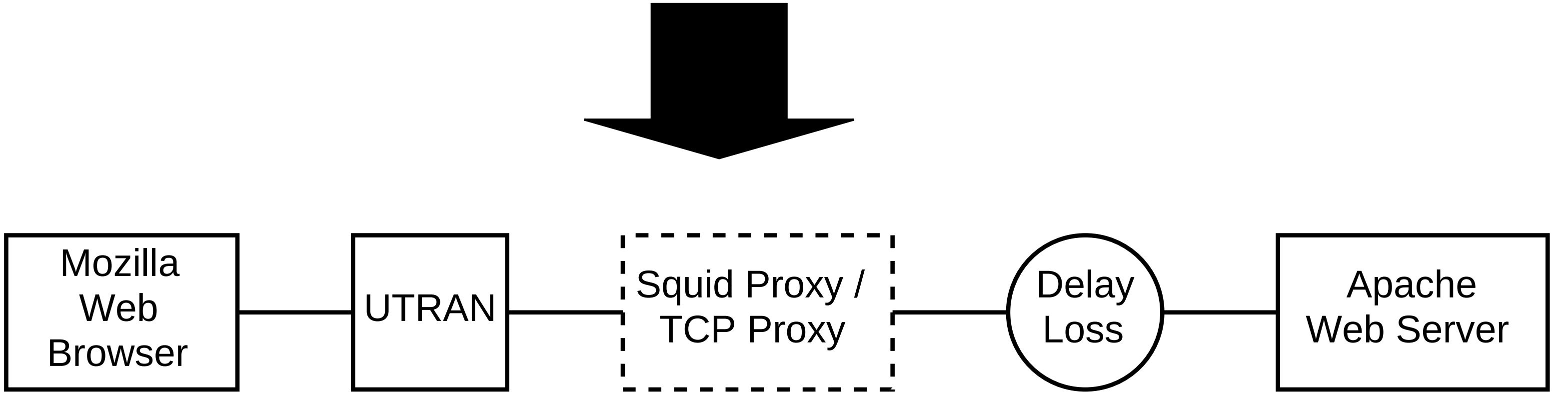
scenarios

- single cell
- single user
- ideal power control

Simlib Usage in Projects

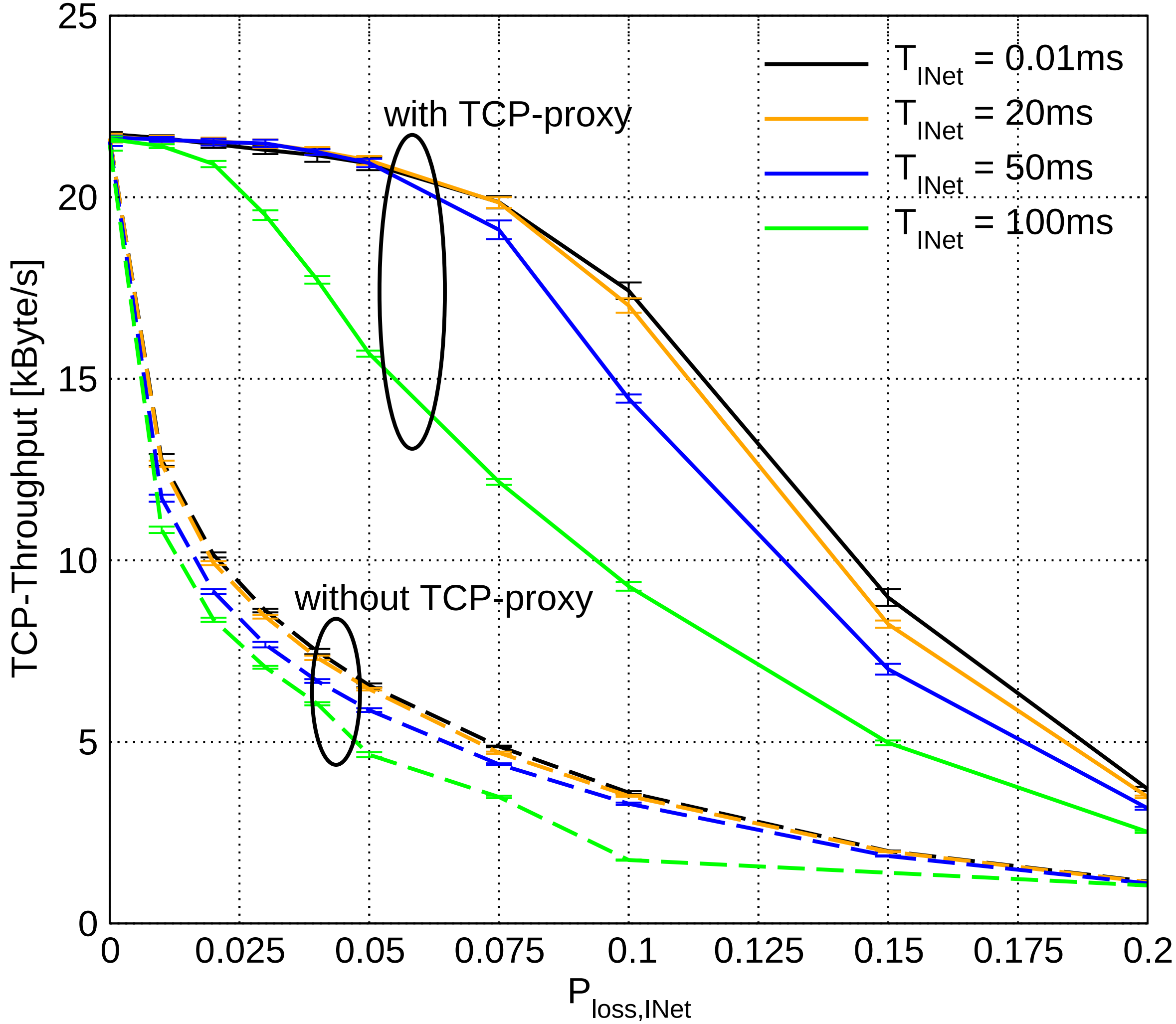
Example 3: Scenario with Proxy





Simlib Usage in Projects

Example 3: Some Simulation Results



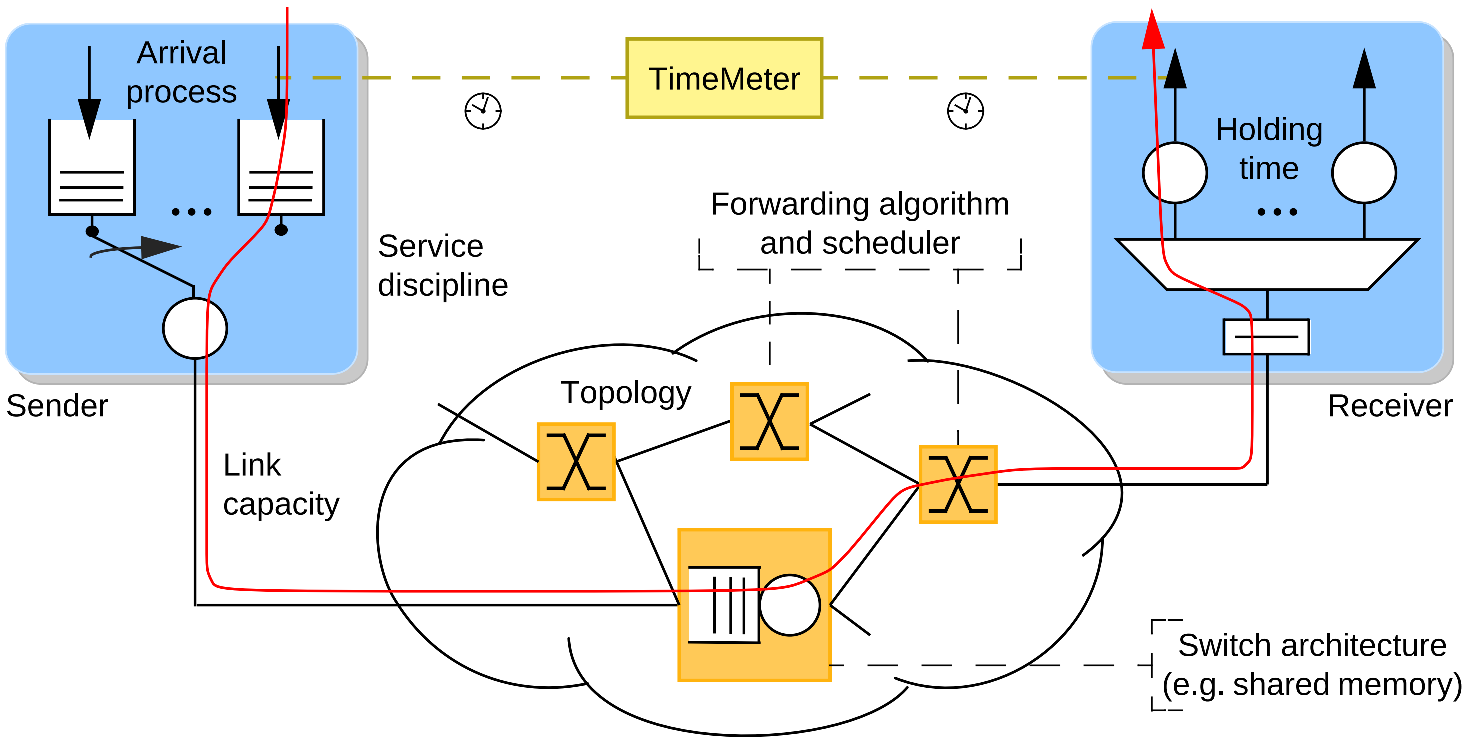
- FTP download
- TCP NewReno
- Downlink
  - DCH 256 kBit/s
  - 20% blocking probability
- Uplink
  - DCH 64 kBit/s
  - 10% blocking probability

- large delays in UMTS slows down recovery from Internet packet loss
- TCP proxy decouples radio access network from Internet

Practical Usage

Example 4: Ethernet networks

- system



- issues
  - system dimensioning (e.g. buffer size, link capacities)
  - latency and jitter (temporal requirements)
  - evaluation of different schedulers
  - frame loss

Further Reading

1. A. M. Law, W. D. Kelton: *Simulation Modeling and Analysis*, McGraw-Hill, 1991.
2. IKR Simulation Library: http://www.ikr.uni-stuttgart.de/IKRSimLib.
3. H. Kocher: *Entwurf und Implementierung einer Simulationsbibliothek unter Anwendung objektorientierter Methoden*, Dissertation, University of Stuttgart, 1994.
4. C. Görg: *Verkehrstheoretische Modelle und stochastische Simulationstechniken zur Leistungsanalyse von Kommunikationsnetzen*, Habilitation, RWTH Aachen, 1997.
5. D. Gross, C. M. Harris: *Fundamentals of Queueing Theory*, 2nd Edition, John Wiley & Sons, 1985.