UNIVERSITÄT OSNABRÜCK

Modeling and Simulation in a nutshell ..

Einführung BA-Seminar

Sommersemester 2019

Prof. Dr. Nils Aschenbruck

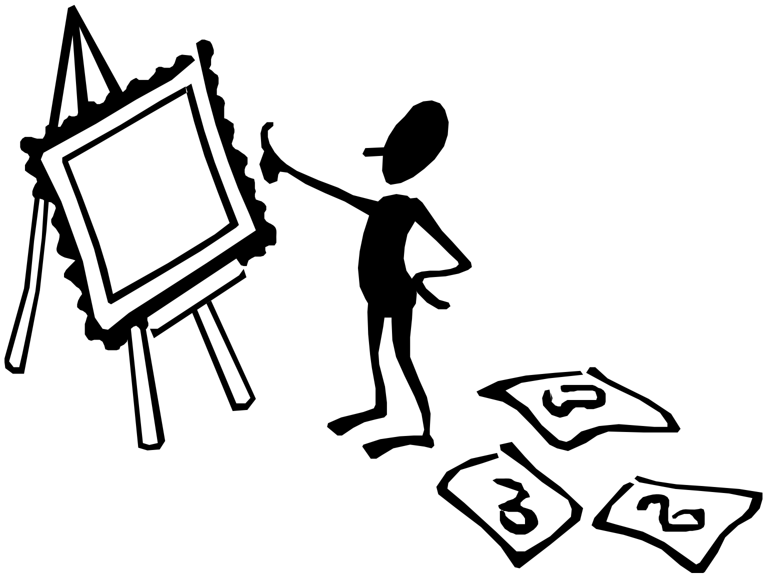
Models should be abstractions capturing relevant aspects of “reality”:

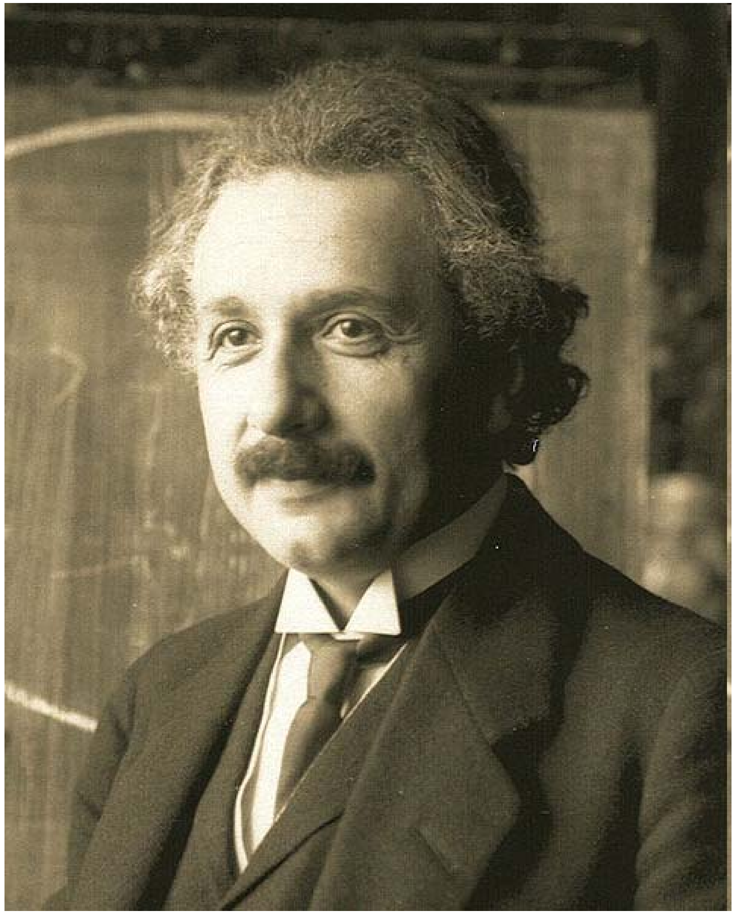
Who does the selection?
Based on which criteria?

Model

An **approximation**, **representation**, **or idealization of selected aspects** of the structure, behaviour, operation, or other characteristics of a real-world process, concept, or system. Note: models may have other models as components

Source: www.sei.cmu.edu/str/indexes/glossary/model.html





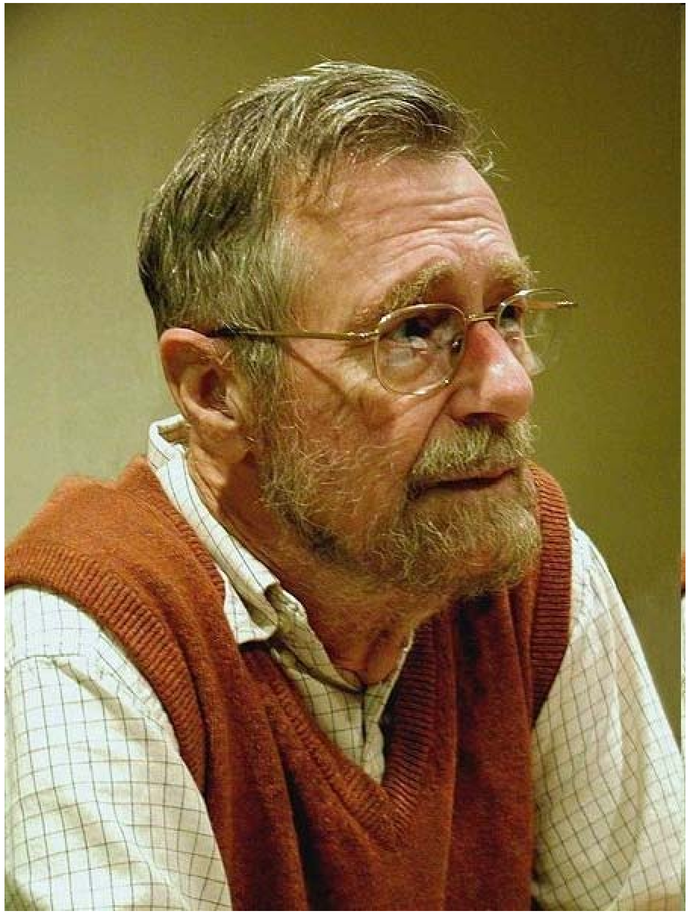
http://en.wikipedia.org/wiki/Image Einstein1921_by_F_Schmutzer_2.jpg

Albert Einstein during a lecture in Vienna, 1921.

"Everything should be as simple as possible but no simpler."

Attributed to Einstein

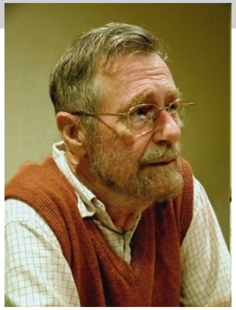
http://en.wikipedia.org/wiki/Image Edsger_Wybe_Dijkstra.jpg



Edsger W. Dijkstra, 1.8.2002

"An abstraction is one thing that represents several things equally well."

Edsger W. Dijkstra



Edsger W. Dijkstra

Two distinct skills are related to abstractions:

1. **Being able to work with a given abstraction**
2. **Being able to develop a useful abstraction**

Dijkstra

Mathematical Methods
Simulation Tools
Measurement Tools

Avoiding assumptions that are not true in reality is
… hard to teach,
… hard to learn,
… but the essence of scientific work.

Simulation allows us to **"play" with mechanisms** in the simulated systems.

Simulation of communication systems is based on

- **protocol specifications** and/or,
- **models of the real world** which (ideally) **include exactly the relevant details** of the selected real world aspects.

"In computer science, simulation refers to the use of computation to **implement a model** of some dynamic system or phenomenon.

The purpose of simulation is usually to make **experimental measurements** or predict behavior, thus **moving the laboratory into the computer** environment.

Simulation thus provides a prototype system with which to answer questions of a **"what if?" nature, or to use for teaching** about the system being simulated."

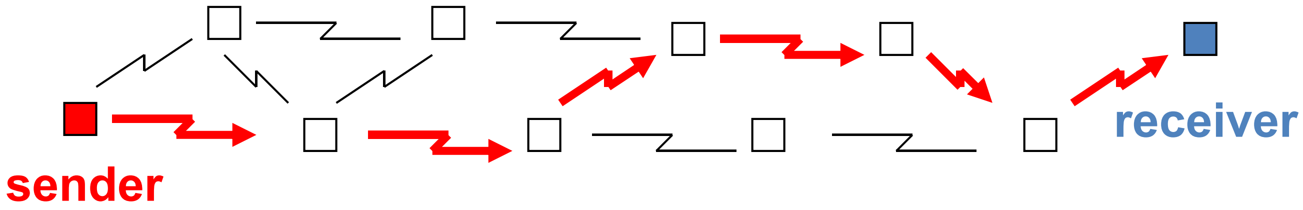
Encyclopedia of Computer Science, IEEE 1993

Note:

In addition to the model-based simulation addressed here, **simulation based on full protocol specifications** is used for **checking functional correctness**.

Tools for this are commercially available, e.g. for SDL.

Your company is developing a new **peer to peer software** for **mobile devices**. The idea is that mobile devices build a **wireless mesh network** in which each device may **share data with all other peers**.



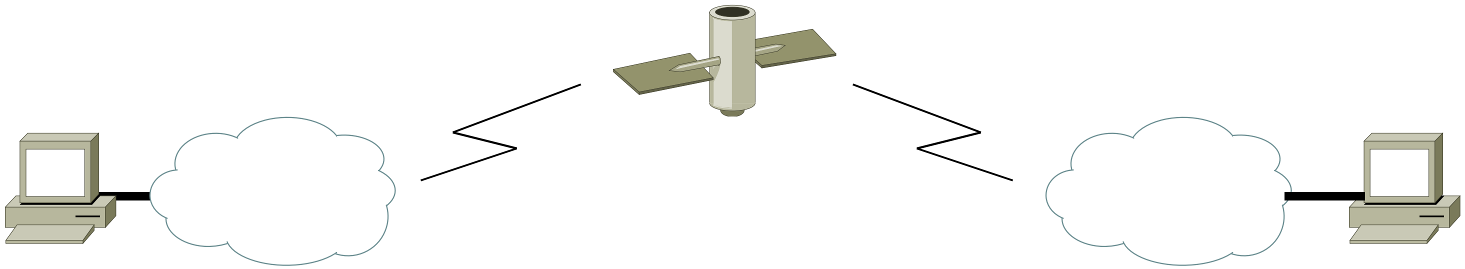
Since the **transmission range** of a single mobile device is **limited**, the system specifies that devices should relay traffic for other devices.

In early development, a **decision has to made between two routing algorithms**.

The system is supposed to scale.

- How to **assess the performance** of each algorithm for 100s or 1000s of nodes?
- Buying as many devices and measuring the performance is **not possible**.

Your company is working on a **"next generation" TCP** that allows efficient data transfer over links with a **high bandwidth delay product** (such as satellite links).



Before deploying the new TCP variant on the hosts of their network, the company has to make sure that it:

- really **performs better than "regular" TCP** in the target scenarios
- **does not misbehave**, e.g. does not push away "regular" TCP traffic (**fairness**)

Real world studies would require an **expensive** and **complex measurement setup**:

- ➢ Expensive satellite links
- ➢ Many physical "regular" TCP senders to examine fairness

How can we obtain the desired information by simpler means?

In network simulations, we usually study **characteristic changes of system state**.

Model World Time: (Simulation Time)	Time in the **simulated system** / in the simulated world.
Real World Time: (Wall Clock Time)	Time in the **real world**, advancing as the simulator executes.

Different approaches to handle changes in the model world time yield different classes of simulation:

Simulation

Continuous-State
(State changing continuously over the [continuous] time)

Discrete-State
(Discrete-Event)

time-driven

(simulation time advances in fixed intervals)

event-driven

(simulation time only changes as events of interest occur)

Fundamental Assumption:

The **state variables are continuous variables.**

Usually, continuous-state simulations **additionally** use **continuous-time models**, i.e. models where the system state changes continually.

Typical Application Areas:

Simulation of physical / chemical phenomena, simulation of streaming behavior, weather forecast, …

Applicability to Communication Systems

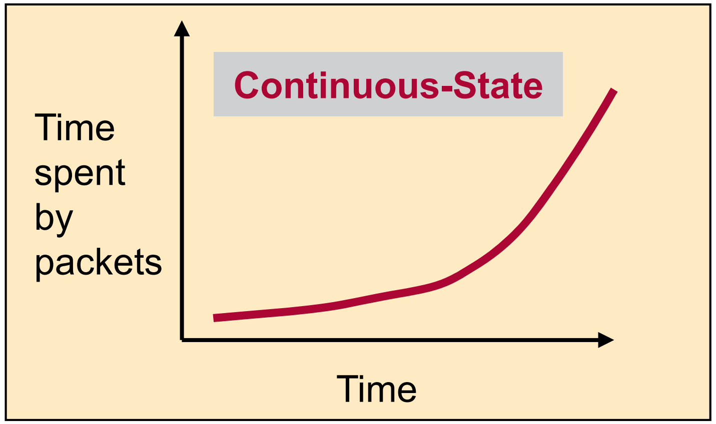
Communication protocols are specified by modified finite state machines consuming **discrete input** (“signals”) such as

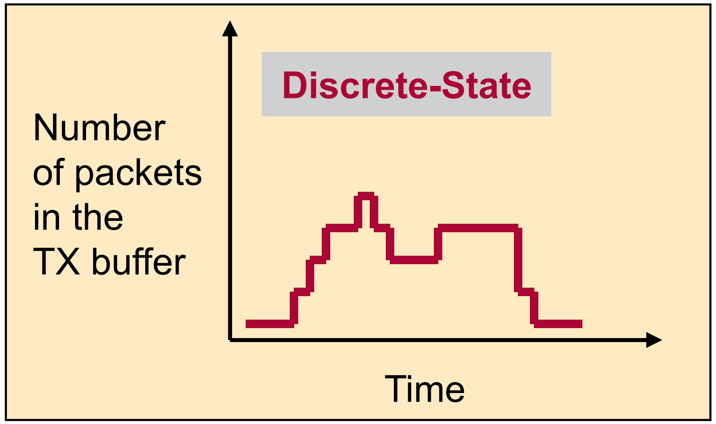
- **message received,**
- **message sent,**
- **timer expired.**

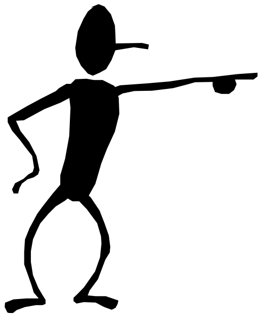
Continuous-state simulation is **rarely used** for communication networks.

Fundamental Assumption:

The **state variables are discrete variables.**







Fundamental Concept of Discrete-State Simulation: The Event

Events change the system state without consuming model world time.

Each event has an associated time value indicating the time when to execute this event.

Examples for Events:

- Enqueueing / dequeueing of messages into/from buffers
- Arrival of a car at a filling station
- Receiving the last and final bit of a message (and thus the complete message)

Discrete-event simulation is extensively used for studying performance aspects of communication systems.

The main component of a discrete-event simulator is a **linked list of the events waiting to happen:**

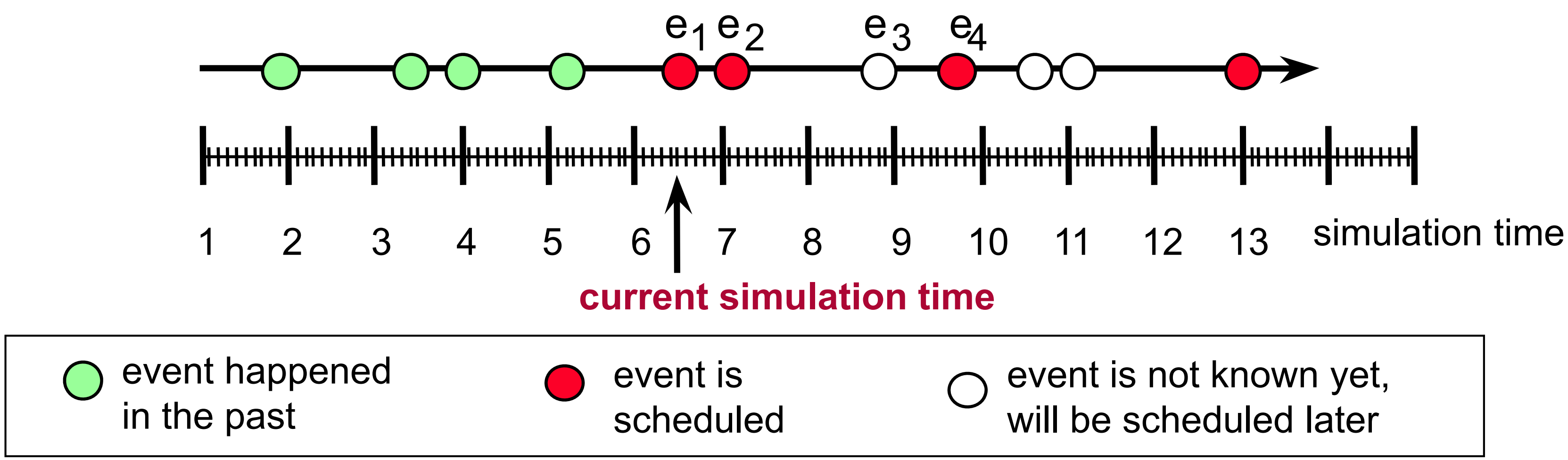
- The **model time** is **advanced from event to event** (to the next earliest event).
- As simulation progresses, events are **added, processed, or dropped.**

Fundamental idea: The model system state only changes with events.

- Define an **initial event**
- **Sort** all future events in a suitable data structure (e.g. priority queue or calendar scheduler)
- **Process** (and then remove) the event with the **earliest event time** by
  - advancing the **model world time** to the event time
  - **invoking the actions** associated with this event.

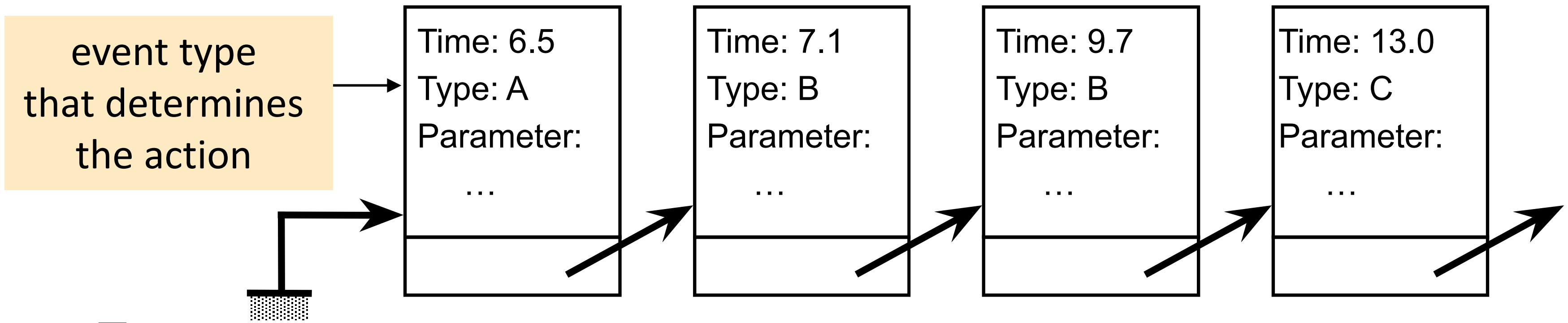
The run time of an event-driven simulation strongly depends on the **number of events** and the **efficiency of event handling**.

In an event-driven simulation, the simulation time has **advanced to time 6.5**.
At this point in time, event ***e_1*** occurs:



Event e_3 may be caused and scheduled by e_1 or e_2.

The event list at simulation time 6.5:

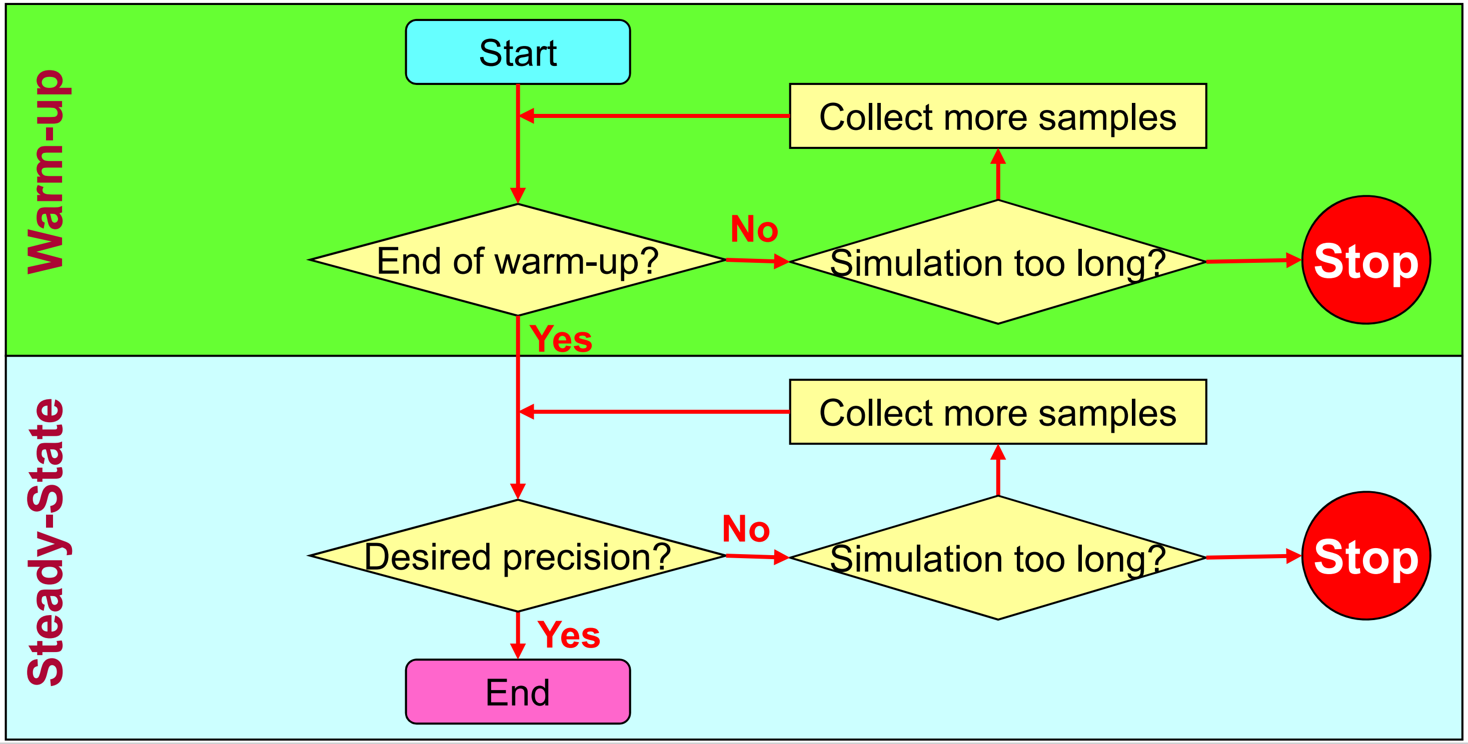


In steady-state simulations, instead of running many replications we

- **collect many samples from a single run**,
- **analyze these samples** for calculating the precision and confidence levels.

Initially, the system remains in a nonstationary **"warm-up period"** where the sample values fluctuate a lot. **Sample values from the warm-up period are discarded** when it comes to calculating the simulation results.

After warm-up, the system becomes stable and asymptotically reaches a statistical equilibrium. The **simulation run time may be based on on-line statistical sample analysis.**

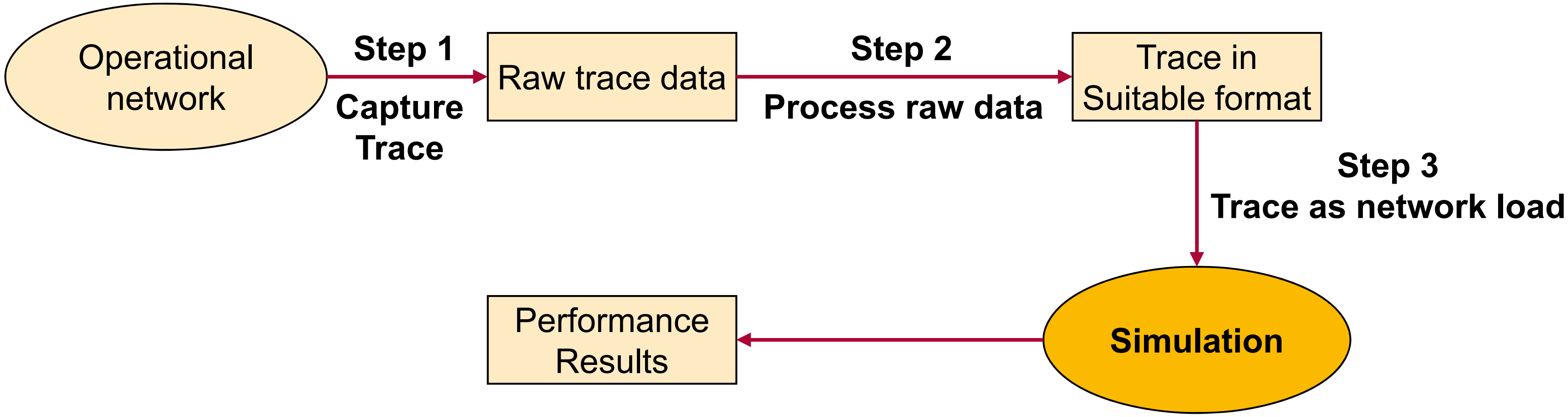


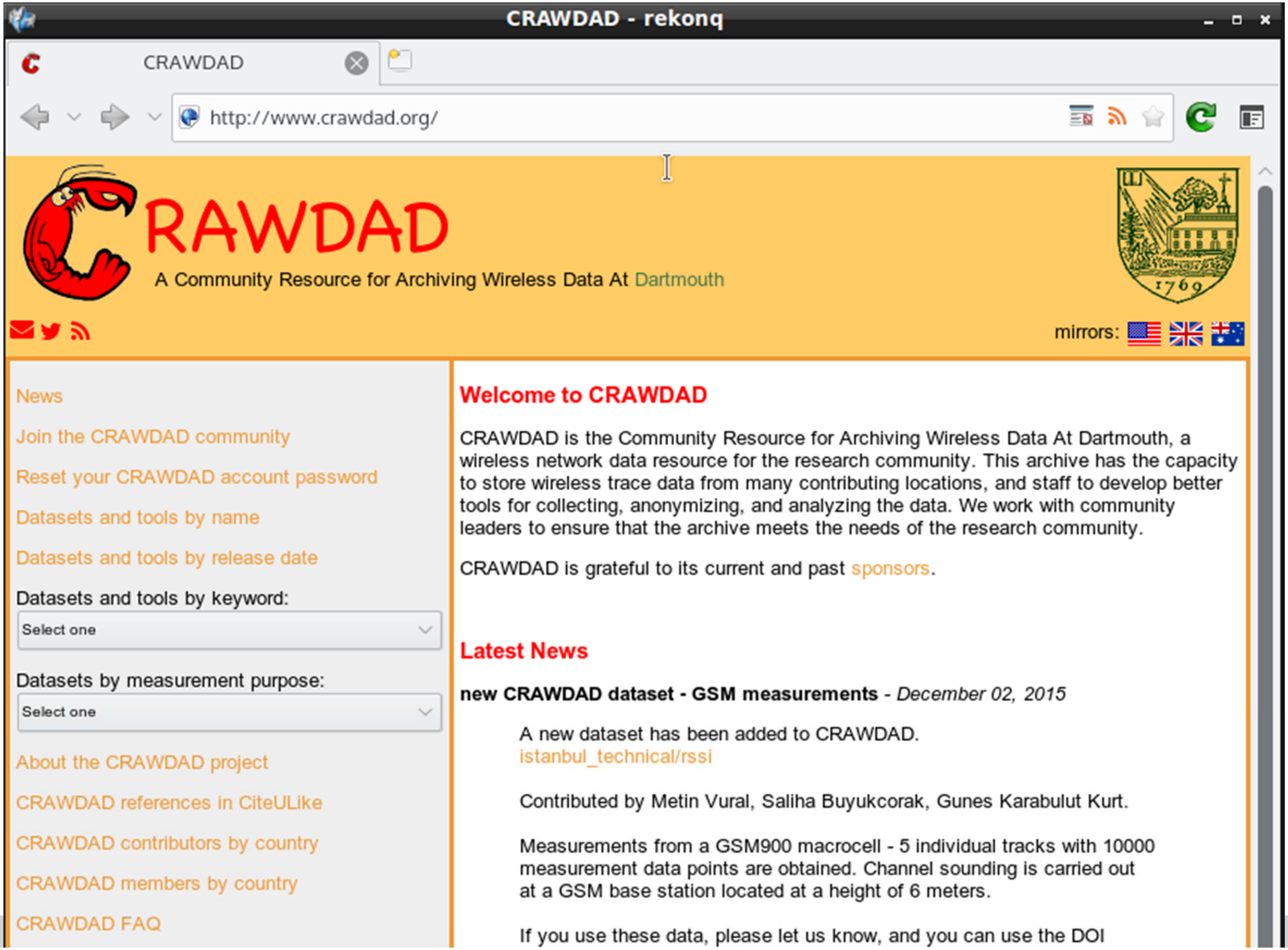
In **synthetic simulations**,

input traffic is synthetically generated by **random load generators** where traffic patterns are chosen from a set of **predefined traffic models** such as Poisson Arrivals, ON-OFF-Sources, self-similar.

With **trace-driven simulations**,

performance analysts **try to reach more credibility** by using traces as input traffic. Traces match 100 % with the actual traffic observed in a specific network during a specific interval of time.




CRAWDAD - rekonq
CRAWDAD
http://www.crawdad.org/
CRAWDAD
A Community Resource for Archiving Wireless Data At Dartmouth
mirrors:
News
Join the CRAWDAD community
Reset your CRAWDAD account password
Datasets and tools by name
Datasets and tools by release date
Datasets and tools by keyword:
Select one
Datasets by measurement purpose:
Select one
About the CRAWDAD project
CRAWDAD references in CiteULike
CRAWDAD contributors by country
CRAWDAD members by country
CRAWDAD FAQ
Welcome to CRAWDAD
CRAWDAD is the Community Resource for Archiving Wireless Data At Dartmouth, a wireless network data resource for the research community. This archive has the capacity to store wireless trace data from many contributing locations, and staff to develop better tools for collecting, anonymizing, and analyzing the data. We work with community leaders to ensure that the archive meets the needs of the research community.
CRAWDAD is grateful to its current and past sponsors.
Latest News
new CRAWDAD dataset - GSM measurements - December 02, 2015
A new dataset has been added to CRAWDAD.
istanbul_technical/rssi
Contributed by Metin Vural, Saliha Buyukcorak, Gunes Karabulut Kurt.
Measurements from a GSM900 macrocell - 5 individual tracks with 10000 measurement data points are obtained. Channel sounding is carried out at a GSM base station located at a height of 6 meters.
If you use these data, please let us know, and you can use the DOI

The most important advantages of simulation:

- **... when compared to mathematical analysis**

  Universal applicability: Simulation may even be used to study scenarios where mathematical analysis fails. However, in these cases the results should be thoroughly scrutinized.

- **... when compared to measurements in real systems**

  Universal applicability: Simulation may even be used to study scenarios where the real system does not exist is too slow, too fast, too dangerous, too expensive ….

Challenges when using simulation:

- **Relevance and credibility of results**

  The quality of simulation results never exceeds the quality of the models used.

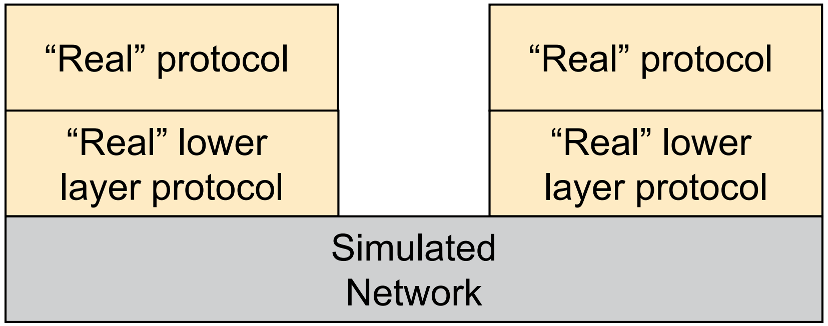
- **Complexity in time and storage space**

  ⇨ … strongly depends on the art of modeling.

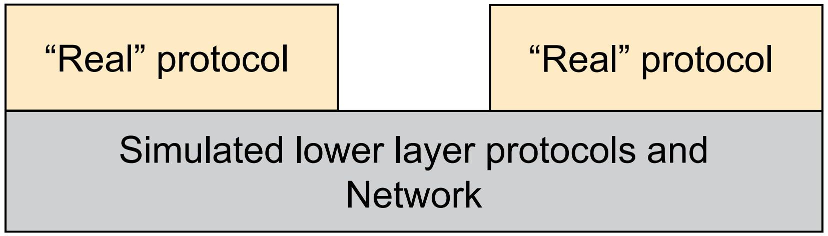
Protocol development for a simulator is **similar** to protocol development for a real system! The only difference is the **interface** to the environment.

Network simulators can be interfaced at different protocol layers:

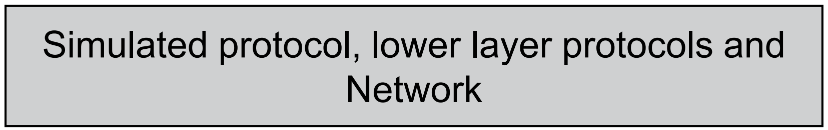
Simulators may only simulate a given **network topology** and use a physical node's real protocol implementations.



Simulators may also provide a **simulation of lower layer protocols** while using a physical node's "real" upper layer protocol implementation.

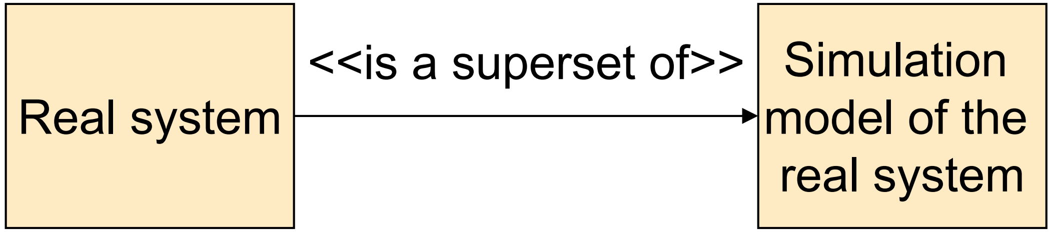


Simulators may also simulate the **full communication system** with all protocols and network nodes.



Create a simulation model of the protocols

- **think** about which characteristics of the system should be reflected in the simulation model
- **implement** the (relevant parts of the) protocols you want to examine
- **avoid unnecessary complexity** in the simulation model



Unnecessary complexity ...

- **does not increase** the accuracy of the results (e.g. you do not need to spend effort in implementing the original TCP header format bit by bit)
- **wastes simulation time** (e.g. you don’t need to simulate an ATM connection with all its protocols when a generic WAN link is sufficient)
- **wastes development time** (e.g. when examining TCP congestion control you don’t need to implement flow control or Nagle’s algorithm)

An idle system is unlikely to show the effects you are interested in…

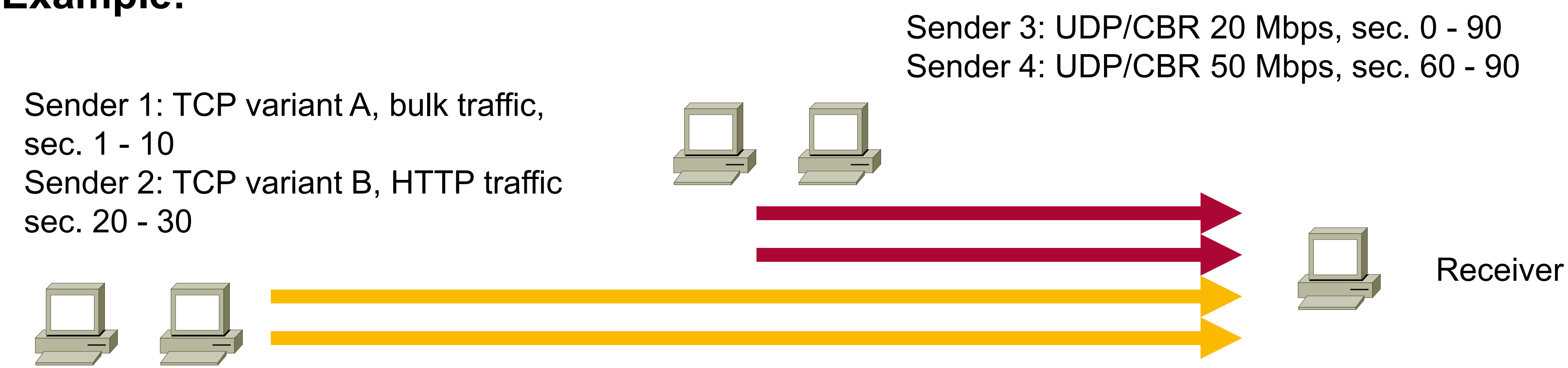
Choose the load model

- simulators often provide a large set of different **load models**
  - **custom load models** (e.g. CBR, Poisson arrivals)
  - **application-specific load models** (e.g. FTP traffic, HTTP traffic, Telnet traffic)
- often, the load model used in a simulation has been **derived from real measurements**

Specify the load model

- write down **what** to send, **when** to send and from **where** to send.

Example:



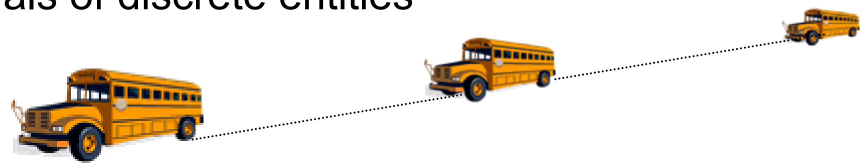
The **performance** of a communication system often depends on the **characteristics of the load** applied to the system.

> Frost/Melamed (1994)
>
> "Traffic modeling is a key element in simulating communication networks. A clear understanding of the nature of traffic in the target system and subsequent selection of an appropriate random traffic model are critical to the success of the modeling enterprise."

Traffic models …

- are a means to **generate load** for measurements, simulation and mathematical analysis.
- are **often derived by measuring real load**.
- can have the form of a **mathematical model** (e.g. expressed in a closed formula)
- can have the form of a **trace file** (e.g. a file that contains the timestamps of packet arrivals)
- are the templates from which to generate **artificial load** that can be **reproduced at any time**, the measurement/simulation should be repeated.

- **simple traffic** consists of simple arrivals of discrete entities
  - busses, trains
  - packets, cells
  - …



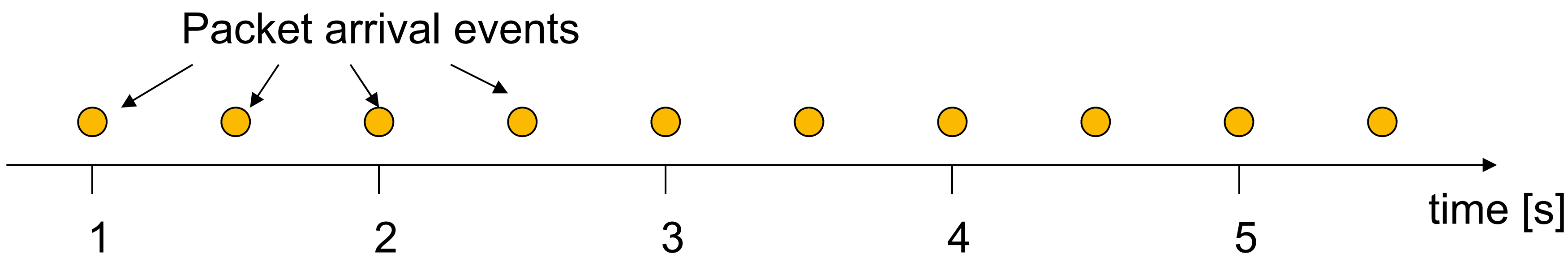
- mathematically described as point process
  - a sequence of arrival instants measured from the origin 0
    - T_1, T_2, T_3, …
    - by convention $T_0 = 0$

- point processes can equally be described as:
  - counting process
    - $\{N(t)\}_{t=0}^{\infty}$ continuous time, non-negative, integer-valued, stochastic process
    - $N(t) = \max\{n : T_n \leq t\}$ is the number of traffic arrivals in (0,t]
  - interarrival time process
    - a non-negative random sequence $\{A_n\}_{n=1}^{\infty}$
    - $A_n = T_n - T_{n-1}$ is the length of the time interval between two arrivals

- **compound traffic** consists of batch arrivals
  - more than one unit arrives at an arrival instant T_n

A simple load model is to generate packets or requests at a constant time interval of length 1 / λ with λ the **arrival rate** (number of packet arrival events per time unit).

The following example shows packet arrival events with rate λ = **2 / s** spaced every 1 / λ = **0.5 s**.



If the packet size is also constant, this load model is called the **constant bitrate** (**CBR**) load model.

CBR model advantages:

- **simple** to implement
- easy to handle in mathematical analysis

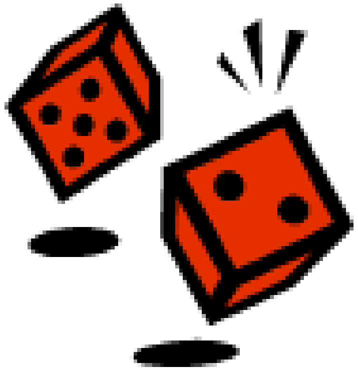
CBR model disadvantages:

- does not match realistic load (however: the **active periods** of **Voice over IP** (**VoIP**) and **video transmission** sometimes appear as CBR traffic)

- time may be
  - slotted (discrete)
    - discrete-time traffic process
  - continuously
    - continuous-time traffic process

- to generate traffic **pseudo-random number streams** are used
  - e.g. generation of inter-arrival times (simple traffic)

    - $T_0 = 0$
    - generate A_1 randomly
    - $T_1 = A_1$
    - $T_{n+1} = T_n + A_{n+1}$



  - compound traffic
    - second stream B_n (e.g., for the packet size / batch size)
    - …

- commonly used:
  - start with one value x_0 the **seed**
  - next numbers in the sequence is a function of previous one(s)

$$x_n = f(x_{n-1}, x_{n-2}, ..)$$

example: $x_n = (5x_{n-1} + 1) \bmod 16$

$x_0 = 5$ => 10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, 14, 7, 4, 5, 10, 3, 0, 1, 6, 15, 12, 13, 2, 11, 8, 9, …

Normalize to [0:1] by div 16:

=> 0.6250, 0.1875, 0.0, 0.0625, …

cycle length is 16

- the sequence depends on the function f and the seed
- f is deterministic => **pseudo-random**
  - preferable for simulation applications
  - to repeat simulation experiments exactly the same way (same sequence)
- pseudo random generators repeat numbers (**cycle length**)

- We need to choose appropriate value for **seed** and appropriate **generator function**.

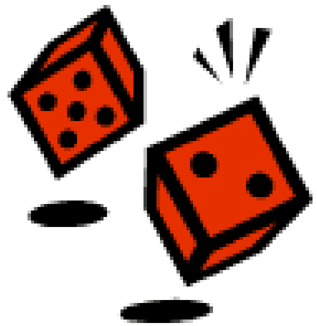
We need to choose appropriate value for **seed** and appropriate **generator function**.

- **generator function:**
  - efficiently computable
  - cycle/period should be large
  - successive values should be independent and uniformly distributed

- **seed: “***any seed value is as good as any other”*
  - do not use zero and avoid even values
    - some RNGs have problems with it
  - do not subdivide one stream
    - multistream simulations
    - to avoid correlations between two variables
  - use non-overlapping streams
    - to avoid correlations between two streams
  - do not use random seeds (e.g., time)
    - simulation can not be reproduced
    - multiple streams may overlap

Non-uniform distributed random variables

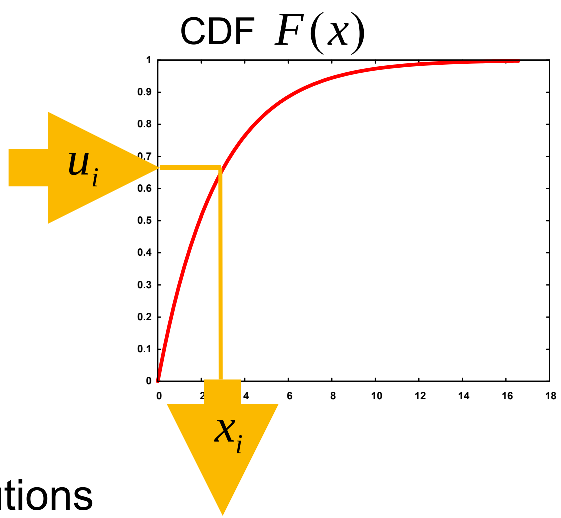
- To generate random variables with a specific random distribution
  - (e.g. exponential, lognormal)



1) obtain a sequence of random numbers
   - uniformly distributed between 0 and 1 $\{u_i\}_{i=1}^{k}$

2) transform it to produce random values $\{x_i\}_{i=1}^{k}$ of the desired distribution with CDF $F(x)$
   - use inverse transformation

$$u_i = F(x_i) => x_i = F^{-1}(u_i)$$



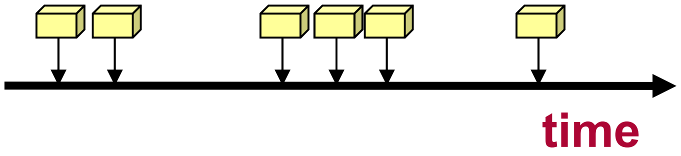
> Example: exponential variates
>
> pdf: $f(x) = \lambda e^{-\lambda x}$
>
> cdf: $F(x) = 1 - e^{-\lambda x} = u \Rightarrow x = -\frac{1}{\lambda}\ln(1-u)$

- further information on generation for specific distributions
  - R. Jain – “The Art of Computer Systems Performance Analysis” – Wiley, 1991 – Chapter 28 and 29

- alternatively use standard libraries
  - GNU Scientific Library (GSL) - http://www.gnu.org/software/gsl/

- due to random number generation and stochastic processes there is no constant arrival
  - packet bursts
  - vehicular traffic bursts

- traffic burstiness is present if arrival points appear to form visual clusters.

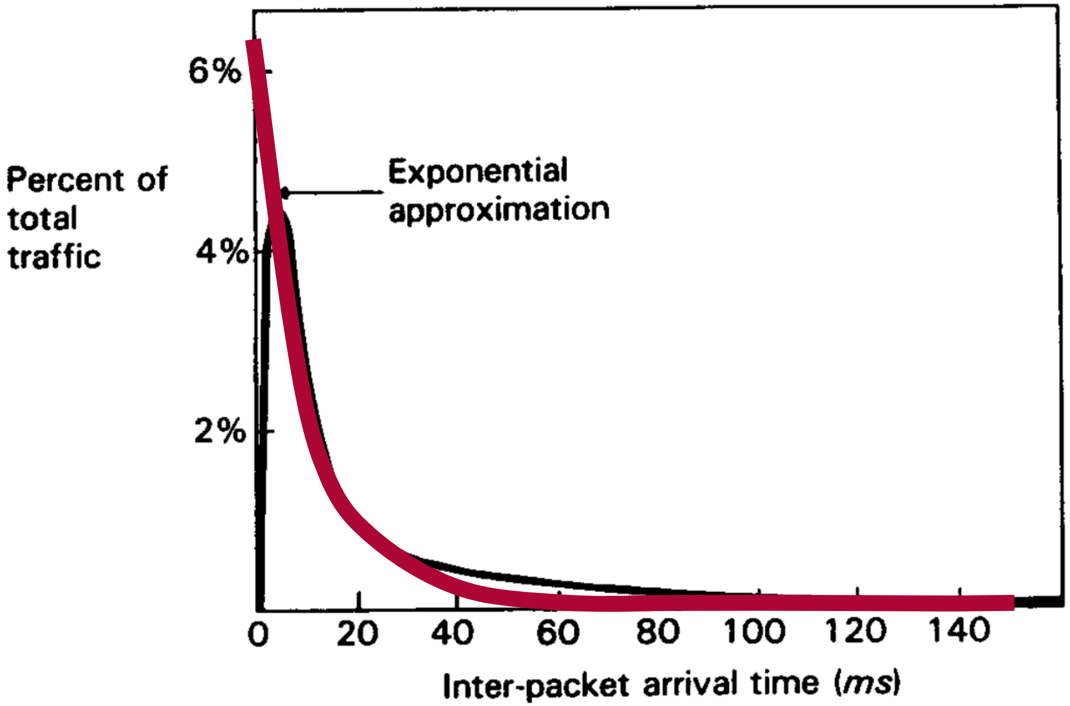


- interarrival times: several short ones are followed by a long one
- reason: short-term autocorrelations of interarrival times

In a renewal traffic process, the interarrival times A_n are **independent, identically distributed**, but their distribution is allowed to be general.

- no dependencies
  - autocorrelation vanishes for all non-zero lags
- memoryless property
- the Poisson process is a renewal process with exponentially distributed interarrival times

An early measurement study of the packet inter-arrival times in an Ethernet LAN revealed the following histogram (Shoch/Hupp, 1980):



This histogram can be approximated by the **probability density function** (**pdf**) of the **exponential distribution.**

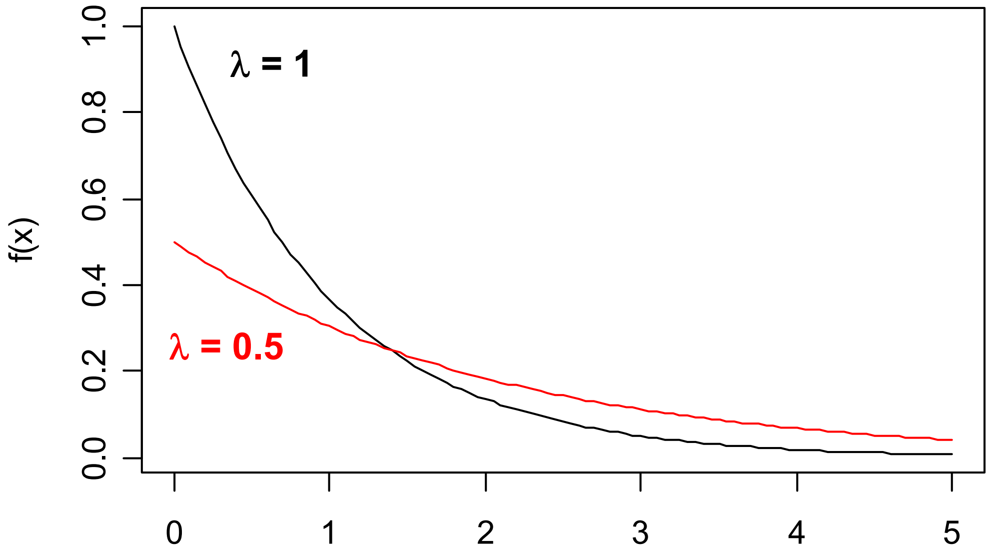
A stochastic process that generates events with exponentially distributed inter-arrival times is called **"Poisson process"**.

If we assume **Poisson Arrivals** then the inter-arrival time of packets is **exponentially distributed**.

The exponential distribution has the **pdf**:

$$f(x) = \lambda\, e^{-\lambda x}$$

where λ **is** the **mean arrival rate**.



Here, the probability of ***k*** arrivals during an interval of length ***T*** [seconds] is given by the **Poisson distribution**:

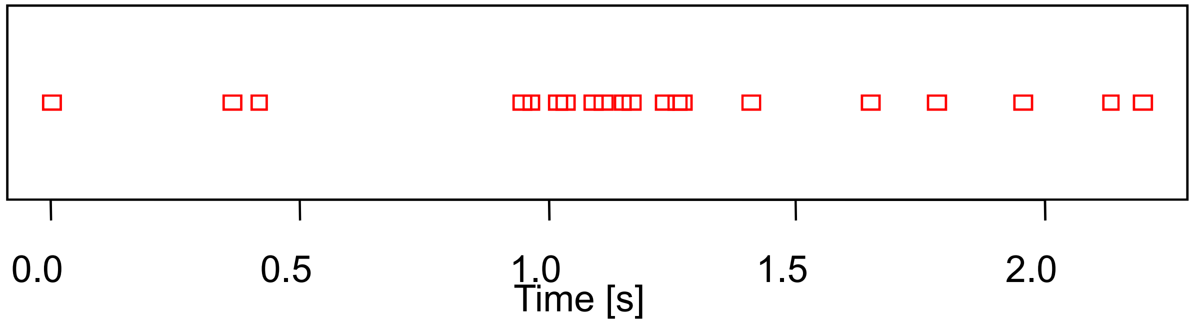
$$P\,\{\,k \text{ arrivals in } T \text{ [seconds] }\} = \frac{(\lambda T)^k\, e^{-\lambda T}}{k!},\ k = 0, 1, 2, \ldots$$

In load generators, exponentially distributed inter-arrival times can be generated by solving the exponential distribution's **cumulative distribution function** (**cdf**) ("Verteilungsfunktion") ***F(t)*** with respect to ***t***:

$$F(t) := \int_0^t f(x)\, dx \;=\; 1 - e^{-\lambda t} =: Y \quad \Leftrightarrow \quad t = -\frac{\ln(1-Y)}{\lambda}$$

For ***Y*** uniformly distributed between 0.0 and 1.0, ***t*** is exponentially distributed with mean λ

Example for exponentially distributed inter-arrival times (λ = **10 / s**):



The arrival times vary and show both, **bursty** and **idle** periods.

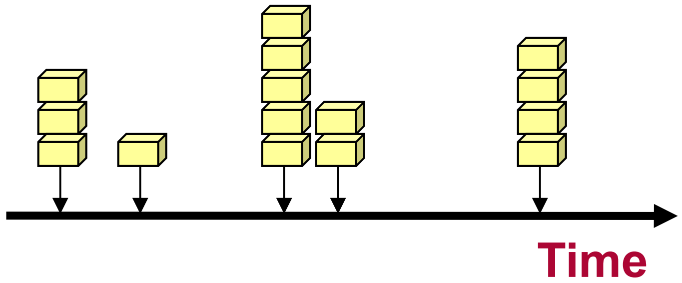
Advantages of Poisson processes for modeling inter-arrival times:

- **“simple“ mathematic analysis**
- superposition of independent Poisson Processes results in a new Poisson Process
  - rate = sum of component rates
- fairly common in traffic applications
  - measurements in **telephone networks** show that Poisson processes are a very close approximation of real arrival processes (e.g. the time between two phone calls can be well approximated by a Poisson process)

Disadvantages of Poisson processes for modeling inter-arrival times:

- Albeit more bursty than CBR, Poisson processes **generate inter-arrival times without excessive peaks**.
- Poisson processes **do not resemble inter-arrival times in current communication networks very closely**.

Real network traffic is more bursty than Poisson traffic. The **Batch Poisson process** models the inter-arrival times exponentially as well. However, instead of single packets, it generates **batches of packets:**



The number of packets in a batch may be modeled by an **arbitrary probability distribution**.

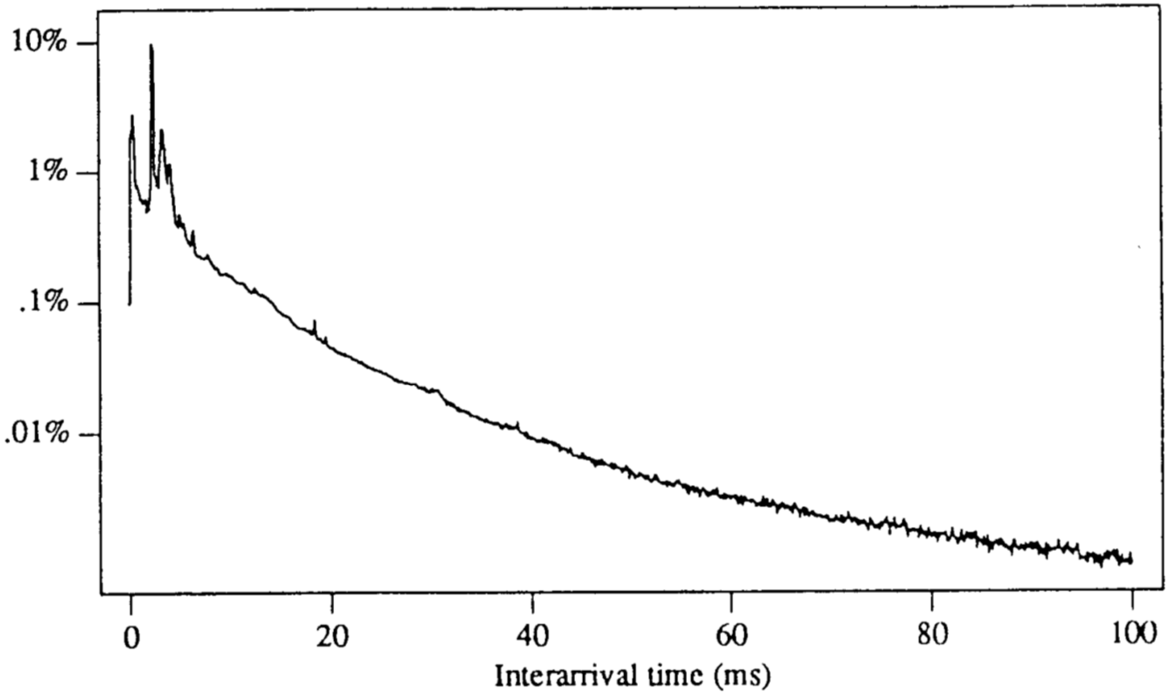
Advantages:

- The traffic generated is more bursty than Poisson traffic.
- **Arrival times are still exponentially distributed** (which simplifies math)

Disadvantage:

- The model **still is not really close to real load**
  - E.g. it ignores the typical **correlation between packet bursts** which is caused by the typical request / response traffic patterns of common communication protocols

Measurements by Gusella at UC Berkeley provided similar results:

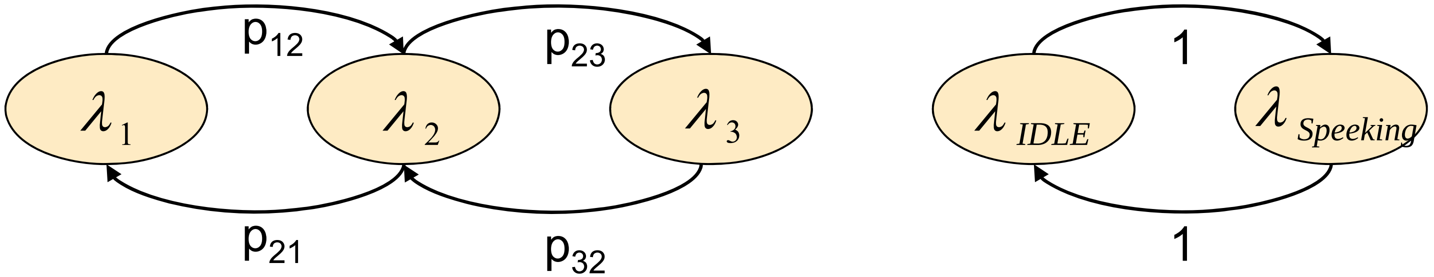


Source:
R. Gusella, “A Measurement Study of Diskless Workstation Traffic on an Ethernet”, IEEE Trans. Commun., Vol. 38, No. 9, Sept. 1990, pp. 1557 - 1568

Note:
The similarity to Poisson Arrivals basically is an illusion:
In real life, there are **strong interdependencies** between successive inter-arrival times (short followed by short, large followed by large).

Markov traffic models introduce **dependence** into the sequence of interarrival times.

- Markov Process:
  - discrete state space
  - exponentially distributed holding times with rate λ_i
  - transition probabilities (Probability matrix) $P = [p_{ij}]$



- next state only depends on current state

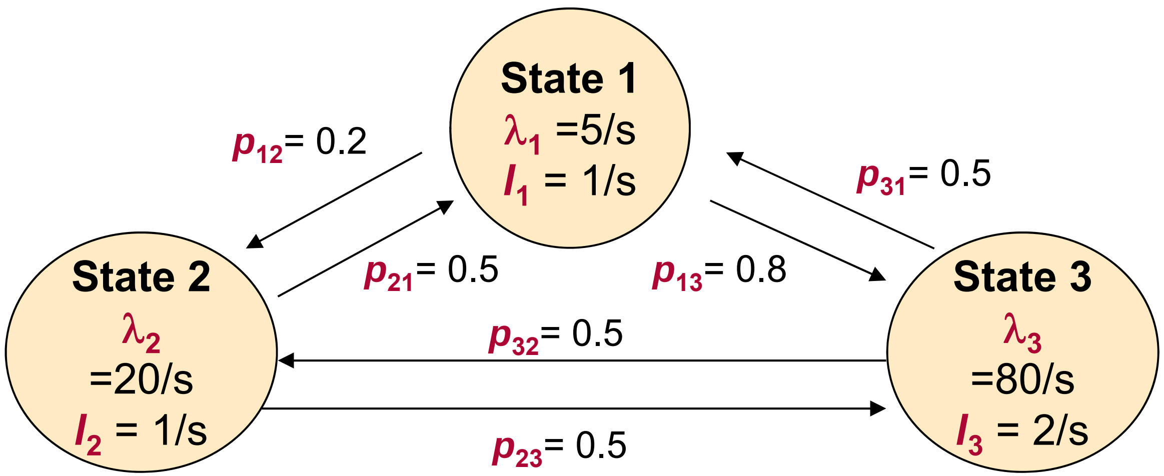
Real network traffic cannot be accurately modeled with a fixed packet arrival rate λ.

A **Markov Modulated Poisson process** (**MMPP**) has multiple states **1**, ..., ***n***. Each state ***i*** is associated two rates λ_i and l_i and a probability vector $\boldsymbol{p}_i = (\boldsymbol{p}_{i1}, ..., \boldsymbol{p}_{in})$

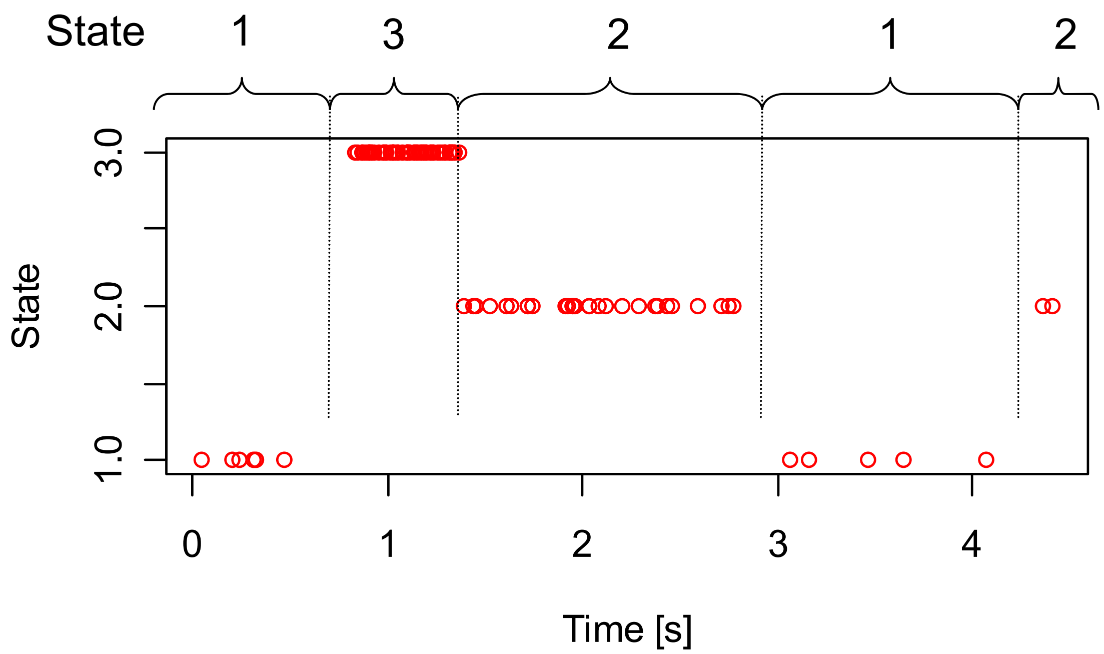
Markov-Modulated Poisson process Traffic Generator

- The process **starts in state 1**.
- The duration, the MMPP stays in state ***i*** is **exponentially distributed** with parameter l_i .
- In state ***i***, the Poisson process is configured with parameter λ_i.
- After that, the process changes to the state ***j*** with probability $\boldsymbol{p}_{ij}$.

Example: A MMPP with three states



Example: Three state MMPP packet arrivals:

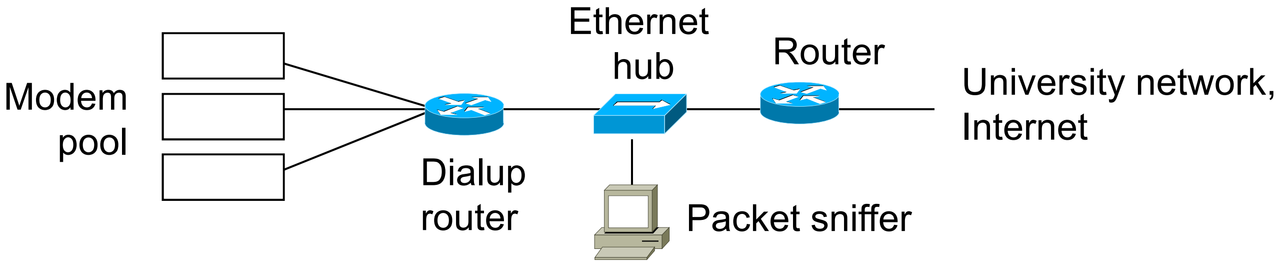


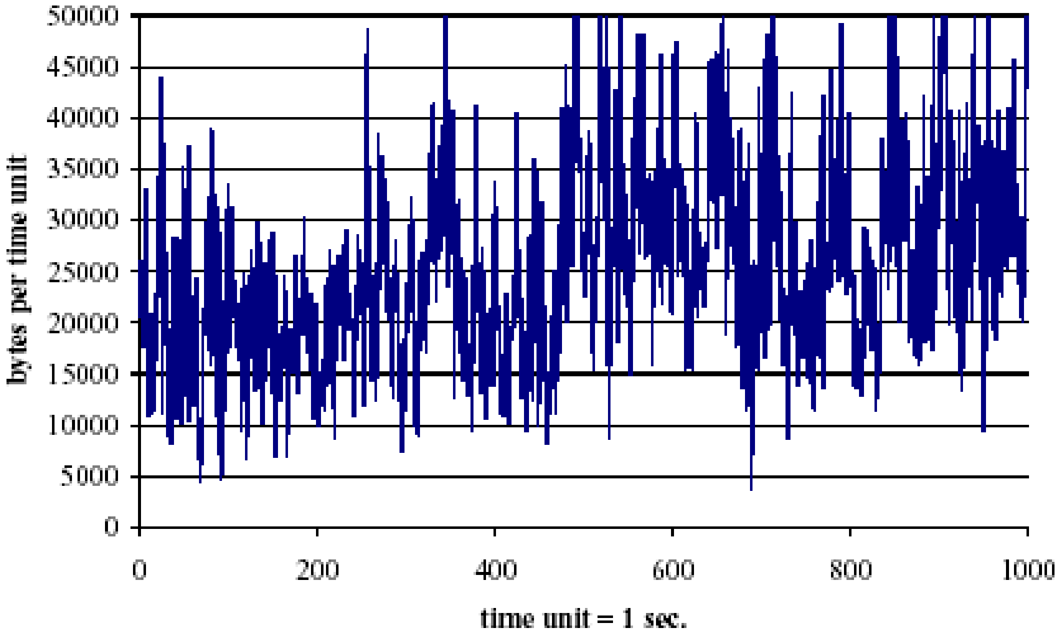
A **MMPP** shows **varying arrival rates** which correspond to its different states.

However, in some scenarios, the load variation is still **not sufficient** to represent real network traffic!

How close are these load models to real network traffic?

The following plot shows a **1000 second slice** of a **throughput measurement** at a ISDN dialup modem pool of the University Dortmund, Germany on Dec. 13, 2000 at 10 a.m.





The traffic shows

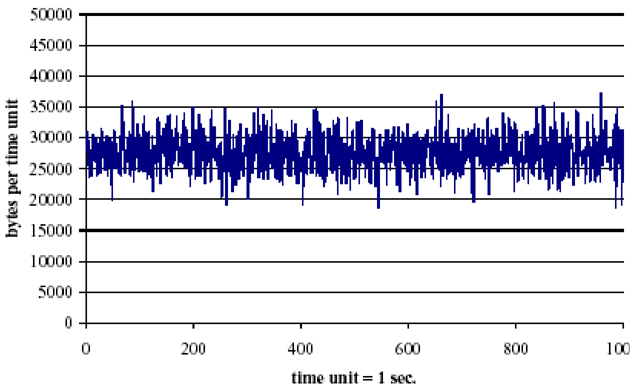
- **severe peaks**
- **short term dynamics**
- **long term dynamics**

A. Klemm, Ch. Lindemann, M. Lohmann, Traffic Modeling of IP Networks Using the Batch Markovian Arrival Process, Proc. TOOLS 2002.

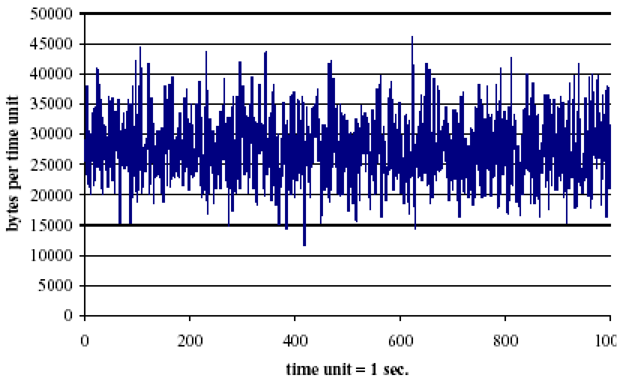
With statistic means, it is possible to derive the parameters of a **Poisson process** and a three state **MMPP** that have similar characteristics than the measured traffic.

The following figures show the "best" attempts to model real traffic with a Poisson process and a MMPP:

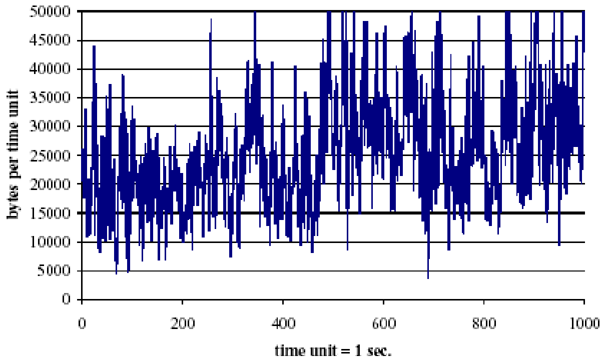
Poisson process



MMPP



Real traffic

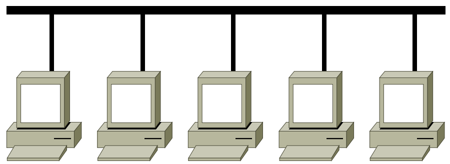


Observations:

- Poisson arrivals are much **smoother than MMPP**
- Real traffic is more **bursty than MMPP**
- The **occurrence of bursts** in real traffic is **bursty**, too!

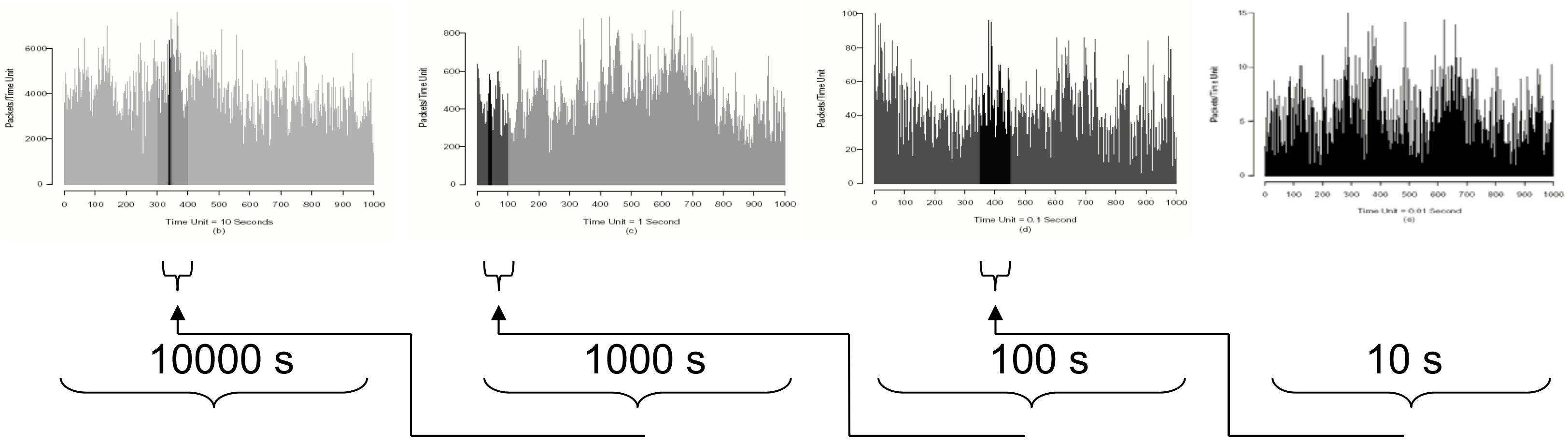
A. Klemm, Ch. Lindemann, M. Lohmann, Traffic Modeling of IP Networks Using the Batch Markovian Arrival Process, Proc. TOOLS 2002.

Leland et al. made clear, that LAN traffic is **self-similar**. Informally speaking, traffic is said to be **self-similar** if it **“looks roughly the same”** regardless at which **time scale** it is observed.



The following throughput plots of Ethernet traffic are based on a single measurement, depicted at **different time scales** from 0.01 s to 10 s.

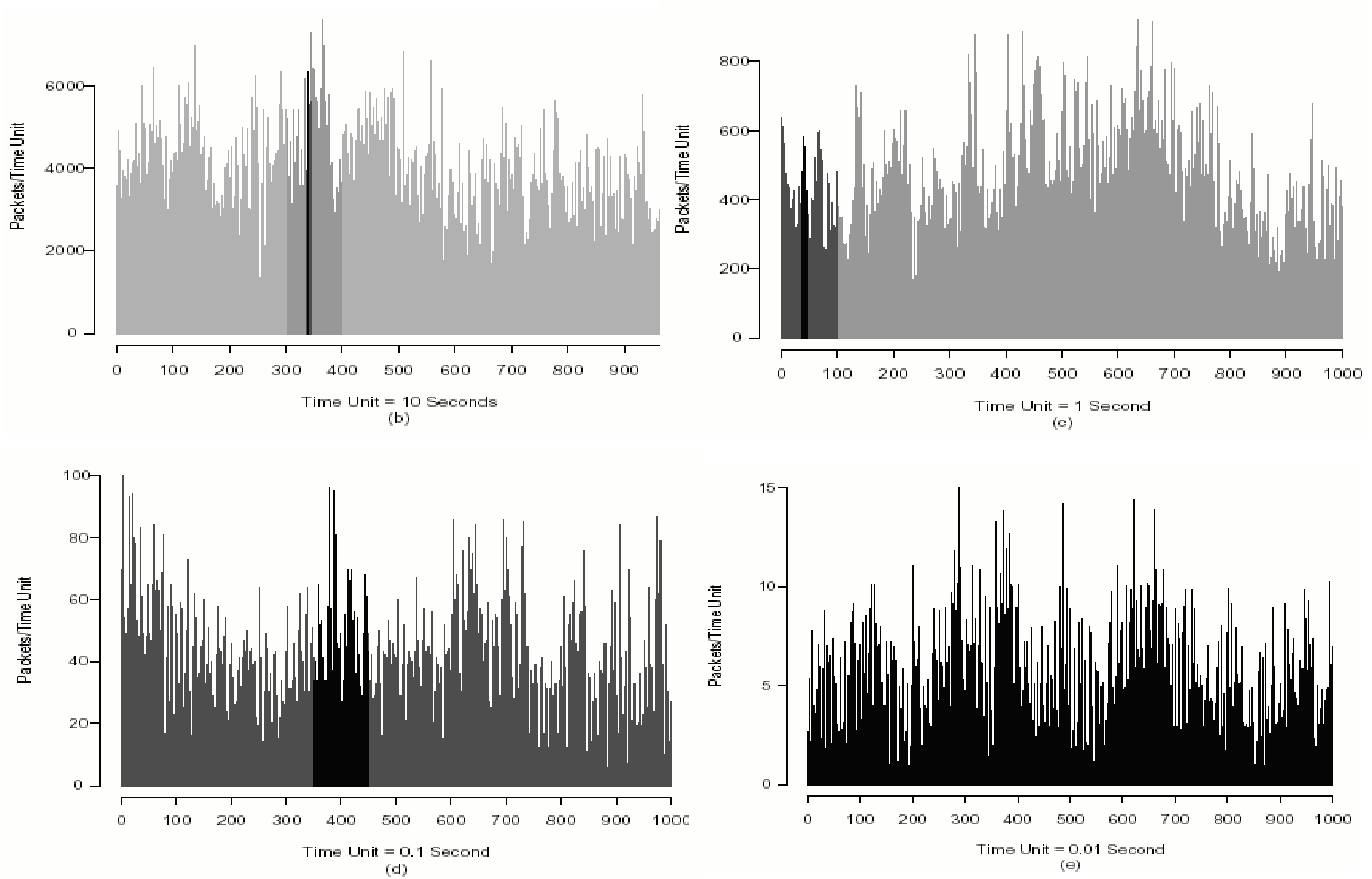
They illustrate that **Ethernet traffic is self-similar**.



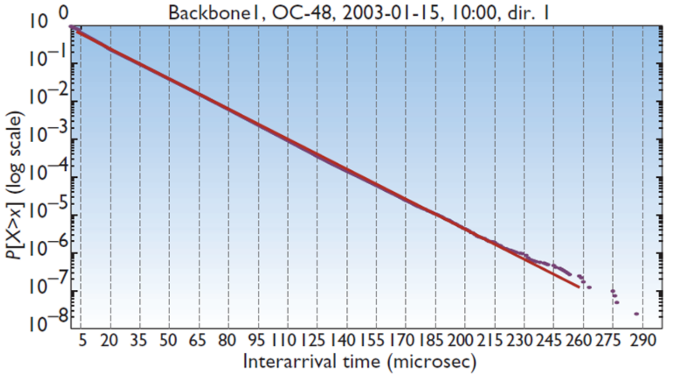
Self-similar load models are an **active area of research**.

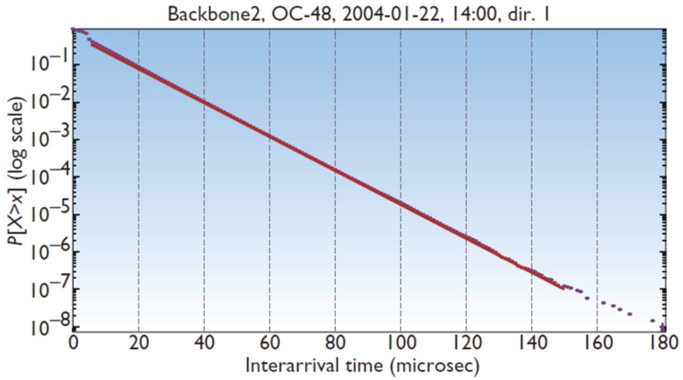
W. Leland et al., “On the self-similar nature of Ethernet traffic (extended version)”, IEEE/ACM Transactions on Networking,Vol. 2, No. 1, Feb. 1994.

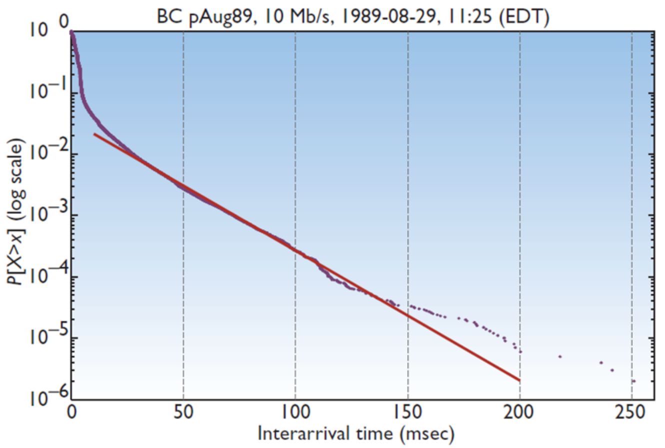
Self-similar Traffic (2)



W. Leland et al., “On the self-similar nature of Ethernet traffic (extended version)”, IEEE/ACM Transactions on Networking, Vol. 2, No. 1, Feb. 1994.







Self-similarity and scaling phenomena have dominated Internet traffic analysis for the past decade. With the identification of long-range dependence (LRD) in network traffic, the research community has undergone a mental shift from Poisson and memory-less processes to LRD and bursty processes. Despite its widespread use, though, LRD analysis **is hindered by the difficulty of actually identifying dependence** and estimating its parameters unambiguously. [..]

[..] We show that unlike the older data sets, **current [backbone] network traffic can be well represented by the Poisson model for sub-second time scales**. At multi-second scales, we find a distinctive piecewise-linear non-stationarity, together with evidence of long-range dependence. [..]

Just as the analogy of the two bugs in the garden shows it is important to **avoid excessively large scales**, we must also be **careful not to focus on too small a time scale**. [..]

T.Karagiannis, M.Molle, M.Faloutsos: “Long-range dependence ten years of Internet traffic modeling”, IEEE Internet Computing, 2004
T.Karagiannis, M.Molle, M.Faloutsos: “A Nonstationary Poisson View of Internet Traffic”, Proc. IEEE Infocom, 2004

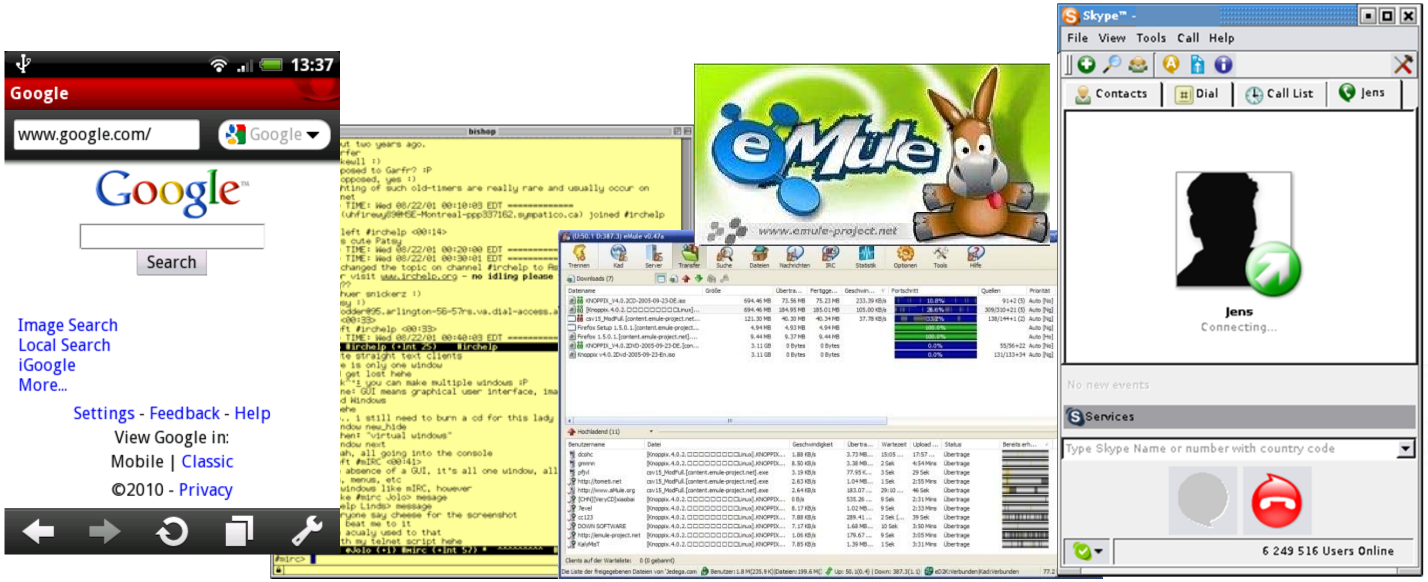
Important: Think of times for single packet delivery (RTTs).

[..] Consider the problem of determining correlation between the motions of **two insects wandering randomly around a small garden**. To an observer in the garden who watches the two bugs, their **motions might appear completely independent and uncorrelated**. However, to an observer **watching the two bugs from outer space**, the motions of the **two bugs appear almost perfectly correlated**, since they are never more than a few inches apart as they traverse a daily rotation of the earth around its axis, which is itself embedded in an annual orbit of the earth around the sun. Clearly, **estimating the motions of the two bugs relative to some "average" derived from celestial-scale measurements is not appropriate for solving this** problem!

Similarly, we should **not try to normalize all network measurements relative to some far away global long-term average value that the system may never reach within the time scales relevant to the calculation of its primary performance metrics**. [..]

T.Karagiannis, M.Molle, M.Faloutsos: "A Nonstationary Poisson View of Internet Traffic", Proc. IEEE Infocom, 2004

- What is realistic data traffic?
- Which kind of traffic is used in the network modeled?
- Which kind of applications are used?



3.8.1 Web-Traffic / HTTP

3.8.2 Voice-Traffic

3.8.3 P2P-Traffic

3.8.4 Chat-Traffic

3.8.5 Video-Traffic

A **mobility model** is designed to describe the movement pattern of mobile users/devices, and how their location, velocity and acceleration change over time.

A **mobility model** is designed to describe the movement pattern of mobile users/devices, and how their location, velocity and acceleration change over time.

Mobility models …

- are a means to **generate movement** for simulation and mathematical analysis.
- can have the form of a **mathematical model** (e.g. expressed in a closed formula)
- can have the form of a **trace file** (e.g. a file that contains certain waypoints)
- are the templates from which to generate **artificial movement** that can be **reproduced at any time**, the measurement/simulation should be repeated.

The **performance** of a wireless communication system often depends on the **characteristics of the movements** of the mobile nodes.

- desireable to emulate the movement pattern of targeted real life applications

  => otherwise simulation results may be misleading

- Trace based models
  - based on user movement traces
  - traces have to be accurate
  - networks have to exist
- Synthetic models
  - capture various characteristics
  - represent movement in a “realistic” fashion
  - macroscopic level
    - previous studies in wireless cellular networks
    - movement of users relative to a particular area (e.g. cell)
      - cell change rate
      - handover traffic
      - blocking probability
  - microscopic level
    - movement of individual nodes
    - node location and relative velocity
    - impact on links

Mobility Model – Classification (1)

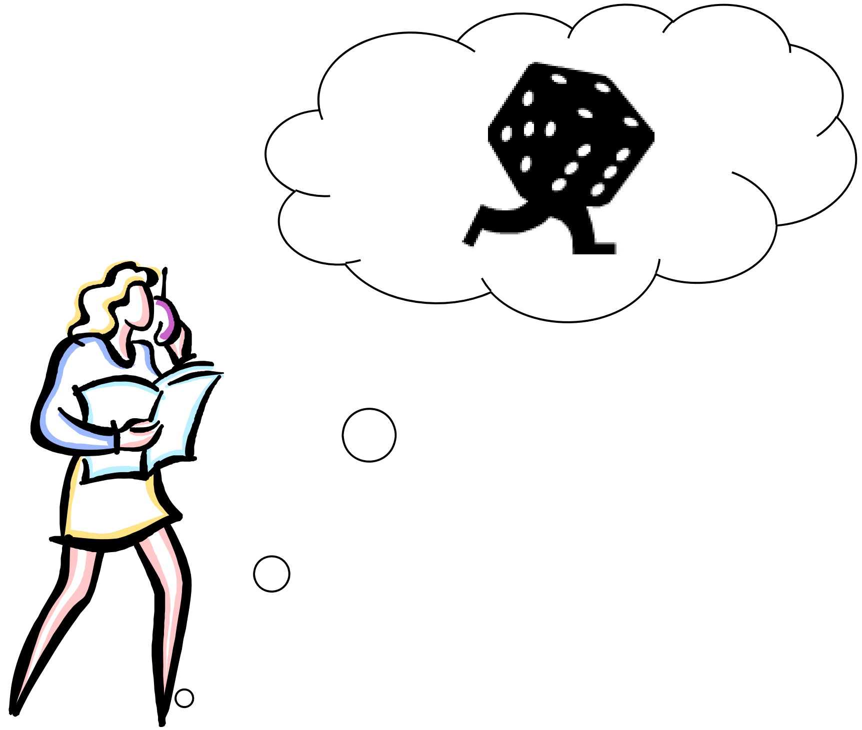
Synthetic mobility models

Random Models

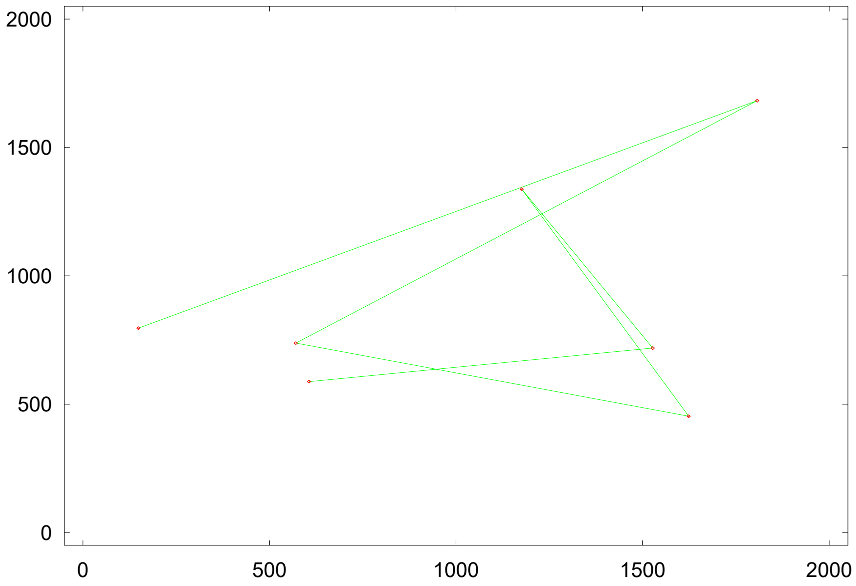
- nodes move to a randomly chosen destination with a randomly selected velocity
- independent of other nodes
- independent of previous movement

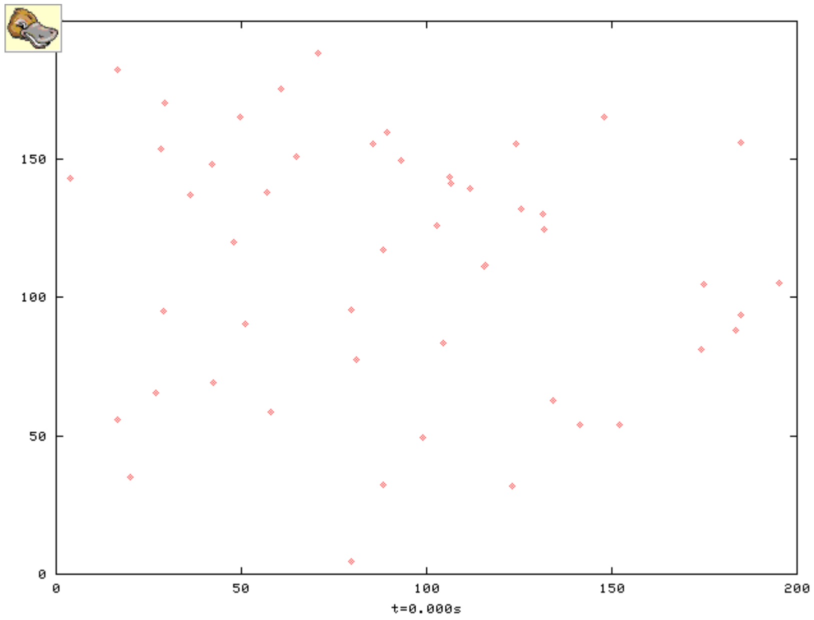
- most widely used
- most intensively analyzed

- Random Waypoint
  - D. Johnson and D. Maltz. - *Dynamic source routing in ad hoc wireless networks* - Mobile Computing, Kluwer Academic Publishers, 1996.
- Random Walk
  - by Einstein 1926



- each node starts at random position
- travels to a randomly chosen destination (x,y)
- with a constant velocity chosen randomly from $[0,V_{max}]$
- bounce if boundary is reached





- often used for performance evaluation (e.g. routing protocols)
  - simplicity
  - easy to implement & widely available
- recently theoretical characteristics are examined using different metrics

Mobility Model – Classification (2)

Synthetic mobility models

- Random Models
- Models with temporal depend.

- nodes move to a randomly chosen destination with a randomly selected velocity
- independent of other nodes
- affected by its movement history

  (speed and direction dependent)

- Gauss-Markov
  - B. Liang and Z. Haas. "*Predictive distance-based mobility management for PCS networks*", Infocom 1999.
- Smooth Random
  - C. Bettstetter. "*Smooth is Better than Sharp: A Random Mobility Model for Simulation of Wireless Networks*", MSWiM 2001

Synthetic mobility models

- Random Models
- Models with temporal depend.
- Models with spatial depend.

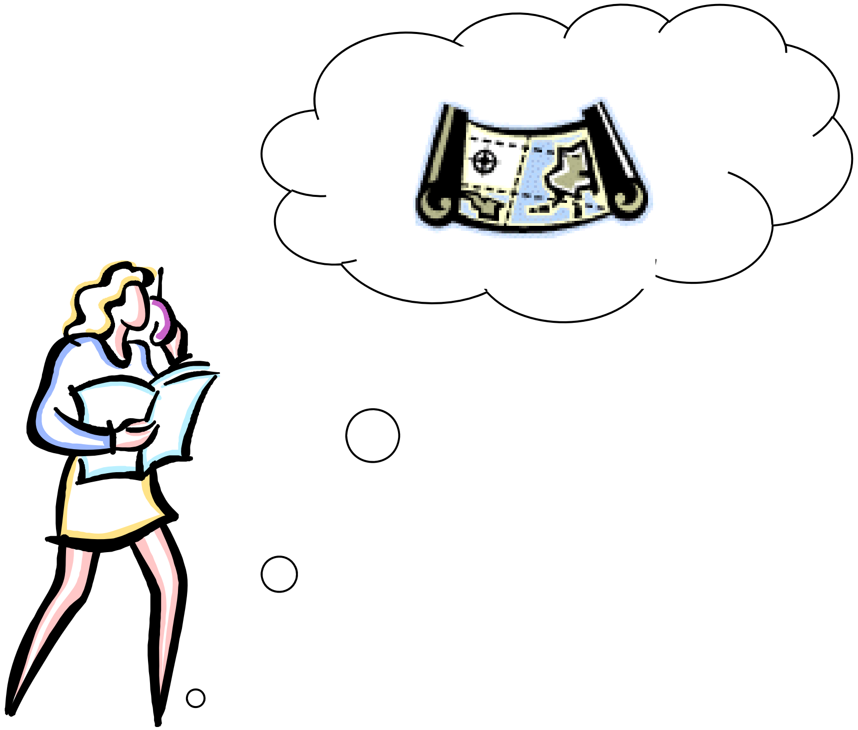
- nodes move to a randomly chosen destination with a randomly selected velocity
- correlated to the movement of other nodes
  - movement in groups
- independent of previous movement

- Reference Point Group Mobility Model
  - X. Hong, M. Gerla, G. Pei, and C. Chiang. "*A group mobility model for ad hoc wireless networks*". MSWiM 1999.
- …

- Synthetic mobility models
  - Random Models
  - Models with temporal depend.
  - Models with spatial depend.
  - Models with geo-graphic restriction

- nodes movement restricted to certain areas
  - streets, freeways, obstacles
- independent of other nodes
- independent of previous movement

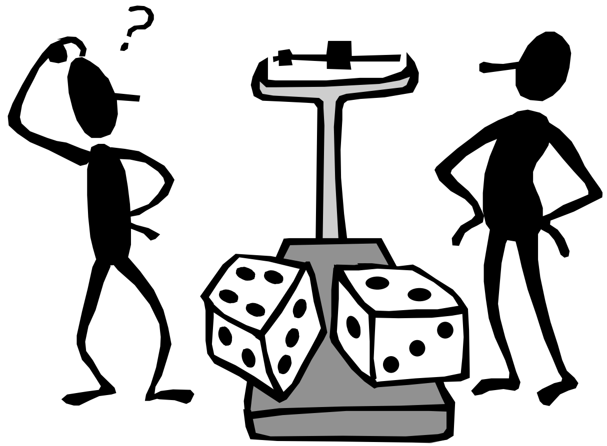
- Manhattan Grid
  - "*Selection procedures for the choice of radio transmission technologies of the UMTS*" - TS30.03 v3.2.0/ 3GPP. 1998
- Obstacle Mobility Model
  - A.Jardosh, E. Belding Royer, K. Almeroth, S. Suri – "Towards Realistic Mobility Models for Mobile Ad hoc Networks" – Mobicom 2003



Model		Dependencies: temporary	spatial	geographical	Applications / Scenarios: Campus	Conference	Pop Concert / Fair	City / Urban	Vehicular	Public Safety	Battlefield	Daily Movement
No dependencies												
Random-Waypoint	[28]											
Random-Direction	[51]											
Modified Random-Direction	[51]											
Random-Walk	[12]											
Random-Border-Model	[8]											
Random-Waypoint with attraction points	[8]						✓					
Clustered-Mobility	[38]		(✓)							✓		
Disaster-Recovery	[46]		(✓)							✓		
General Ripple	[13]											
Temporary dependencies												
Gauss-Markov	[35]	✓										
Smooth-Random	[6]	✓										
Spatial dependencies												
Reference-Point-Group	[21]	(✓)	✓	(✓)								
Structured-Group	[10]		✓							(✓)	(✓)	
Virtual Track	[59]		✓					✓			✓	
Social-Network-founded	[42]		✓									
Mold	[36]		✓		✓							
Community-based	[43]		✓									
Geographic restrictions												
Manhattan-Grid	[17]			✓				✓				
Graph-based	[54]			✓				✓				
Obstacle	[26]			✓	✓							
Weighted-Waypoint	[23]			✓	✓							
Voronoi	[60]			✓				✓				
Area-Graph-based	[9]			✓	✓							
CosMos	[19]			✓								
Hotspot	[39]			✓			✓					
Route	[39]			✓				✓				
Random-Waypoint-City	[30]			✓				✓	✓			
Agenda Based	[58]			✓				✓				✓
Graph-Random-Waypoint	[40]			✓				✓	✓			
Graph-Random-Walk	[40]			✓				✓	✓			
Subway	[55]			✓				(✓)				
Hybrid dependencies/restrictions												
Freeway	[4]	✓	✓	✓					✓			
User-oriented-Meta-Model	[53]	✓	✓	✓								✓
Street-Random-Waypoint	[14]	✓	✓	✓				✓	✓			
VanetMobiSim	[20]	✓	✓	✓				✓	✓			
Hostage-Rescue	[25]		✓	✓							✓	
Disaster-Area-Model	[2]		✓	✓						✓		
CORPS	[24]		✓	✓						✓		
Platoon	[49]		✓	✓							✓	
Working-Day-Model	[16]	✓	✓	✓				✓				✓

Source: Nils Aschenbruck, Aarti Munjal, Tracy Camp
"Trace-based Mobility Modeling for Multi-hop Wireless Networks"
Elsevier Computer Communications, Volume 34, Issue 6, May 2011, pp. 704-714.

- Why to use mobility metrics?
  - mobility metrics are needed to perform statistical analysis of the generated movement
  - statistical analysis enables us to figure the characteristics of the generated movement
  - only if we know these characteristics, we can properly interpret the results of the performance evaluation



- mobility metrics
  - protocol independent metrics
  - protocol performance metrics

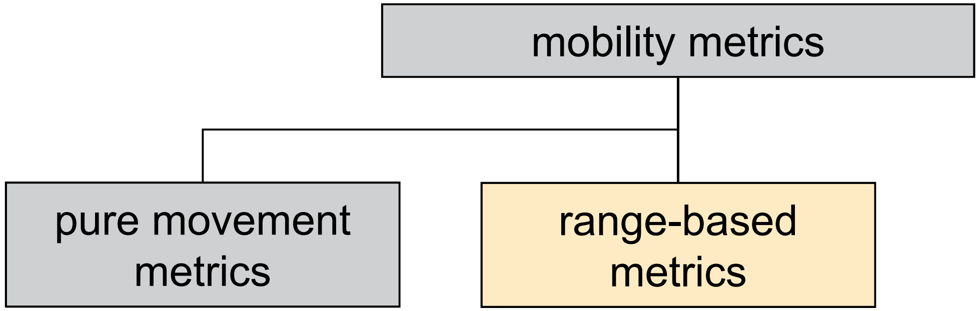
(Source: F. Bai, N. Sadagopan, and A. Helmy, "*IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of RouTing protocols for Adhoc NeTworks*" Proc. of the IEEE Infocom,2003) mobility metrics

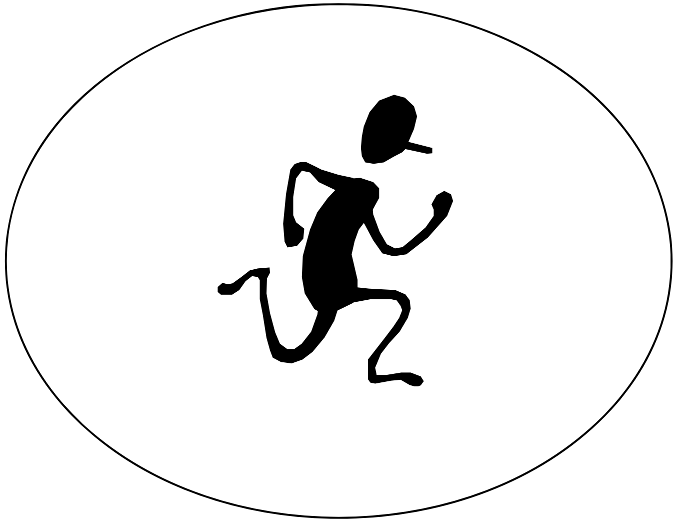
pure movement metrics

- no further assumptions
- are calculated from node statistics

- average velocity
- relative mobility
  - P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark – “*Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks*” - Proc. of the IEEE Mobicom, 1999.
- node distribution
  - C. Bettstetter and C. Wagner – “*The Spatial Node Distribution of the Random Waypoint Mobility Model*” - Proc. of the 1st German Workshop on Mobile Ad-Hoc Networks (WMAN02), 2002.

Metrics for Mobility



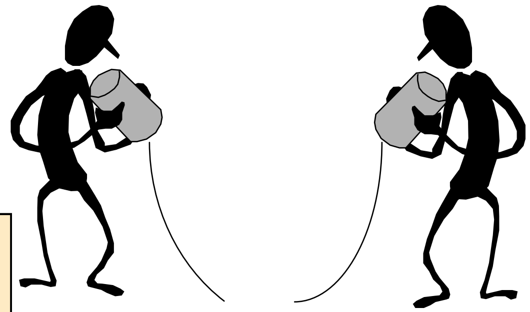


- contain a range around the node as one parameter
  - (e.g. transmission range or communication range)
- the choice of the range parameter is not trivial
- often assume that far away nodes have less impact than nodes nearby

- concept of remoteness
  - B. Kwak, N. Song, and L. Miller – “*A canonical measure of mobility for mobile ad hoc networks*” - Proc. of the IEEE Milcom, 2003.
- degree of spatial dependence
  - F. Bai, N. Sadagopan, and A. Helmy, “*IMPORTANT: A framework to systematically analyze the Impact of Mobility on Performance of RouTing protocols for Adhoc NeTworks*” Proc. of the IEEE Infocom, 2003

Metrics for Mobility

- mobility metrics
  - pure movement metrics
  - range-based metrics
  - link-based metrics



- depend on links between two nodes
- whether a link exists or not depends on the propagation model assumed
- for simplicity reasons, often times a circular range around a node is assumed

- average node degree
- average link duration
- average time to link break
- number of partitions / partition degree

- A. Allen – “Probability, Statistics, and Queueing Theory with Computer Science Applications” – 2nd ed. – Academic Press, 1990
- R. Jain – “The Art of Computer Systems Performance Analysis” – Wiley, 1991
- A. Law, W. Kelton – “Simulation Modeling and Analysis” – 3rd ed. – McGraw-Hill, 2000
- S. Ross – “Introduction to Probability and Statistics for Engineers and Scientists” – 3rd ed. – Elsevier, 2004