SZÉCHENYI EGYETEM
UNIVERSITY OF GYŐR

Doctoral School of Multidisciplinary Engineering Sciences (MMTDI)

Performance Analysis of ICT Systems

Lecture 1

Introduction to the methods for performance analysis of ICT systems and presentation of results

https://www.tilb.sze.hu/cgi-bin/tilb.cgi?0=m&1=targyak&2=NGD_MDA64_1

Dr. Gábor Lencse

Professor

Dept. of Telecommunications, University of Győr

lencse@sze.hu

Agenda

- Reminder: ICT systems
- Introduction to the methods for performance analysis of ICT systems.
- The fundamentals and terminology of modelling and simulation.
- Methods and tricks for presenting results.

Legacy ICT systems

- Standalone systems, dedicated for a single purpose
  - PSTN (analogue, digital), ISDN
  - Cellular phone systems (GSM in Europe, CDMA in the USA)
  - Digital Trunked Radio Systems, e.g. TETRA
  - Analogue/Digital radio broadcasting systems
  - Analogue/Digital video broadcasting (DVB-S/C/T)

IP-based ICT systems

- Common part: Internet Protocol version 4 or 6
- Several solutions to carry the IP datagrams
- A few higher layer protocols over IP
- A high number of application on the top

IP-based ICT systems

- Physical / Data link layer implementations to carry IP datagrams:
  - LAN (Ethernet)
  - Wireless LAN (IEEE 802.11, 11a/b/g/n/ac)
  - [PAN (802.15)]
  - Wired access (ADSL, DOCSIS, FTTx)
  - Wireless access (GPRS, UMTS, HSDPA/HSUPA, LTE)
  - WAN (X.25, frame-relay)
  - SDH (as long distance carrier of IP)

IP-based ICT systems

- Most important protocols
  - TCP/IP protocol stack
    - IPv4/IPv6, ICMP/ICMPv6, TCP, UDP, IGMP/MLD
  - Routing protocols
    - Unicast routing: RIP, OSPF, BGP
    - Multicast routing: PIM-SM/DM, MOSPF, DVMRP
  - QoS Methods
    - IntServ, DiffServ, MPLS
  - Higher layer protocols
    - RTP, RTCP; RTSP

IP-based ICT systems

- Network Applications
  - DNS
  - SMTP, POP3/POP3S, IMAP4/IMAP4S,
  - FTP, NFS, SMB, HTTP/HTTPS,
  - TELNET, SSH, SCP
  - SIP, H.323
  - IPTV
  - BitTorrent

Typical questions

- Many times not really technical, sometimes vague and not clear…
  - What are the limits of the system?
  - What performance reserve does the system have?
  - What will be the delay and jitter, if…?
  - How much extra traffic may be allowed to the system, while maintaining certain QoS parameters?
  - Where are the bottlenecks?
  - What kind of resources should be added?

Typical questions

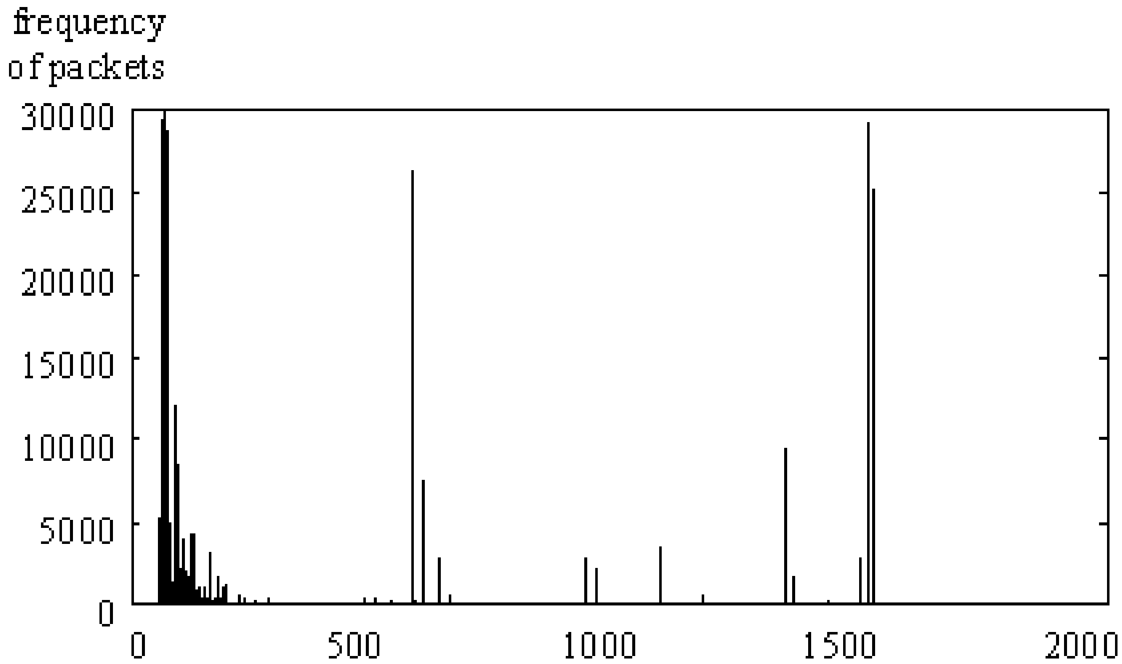
- Many times not really technical, sometimes vague and not clear…
  - After adding given resources, how certain performance characteristics will change?
  - Based on the current trends, what can we expect at a later point of time?

Methods of Performance Analysis

- Measurements (taken on a real system)
- Analytical method (mathematical examination)
- Simulation (experimentation on a computer executable model of a system)

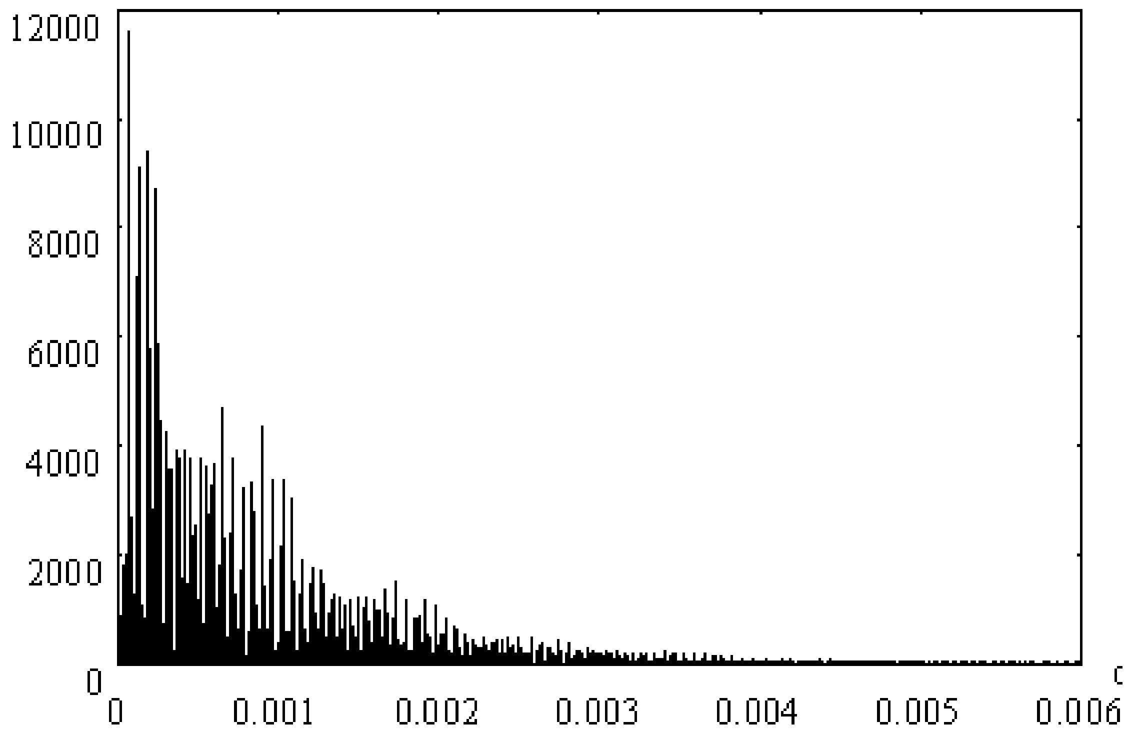
Measurements

- Can give the most accurate and reliable results
- Very important, necessary for the other two methods, too
- Examples:



Packet length statistics measured on the FDDI backbone of BUTE in 1996.

Measurements



*Inter-arrival time statistics** measured on the FDDI backbone of BUTE in 1996.

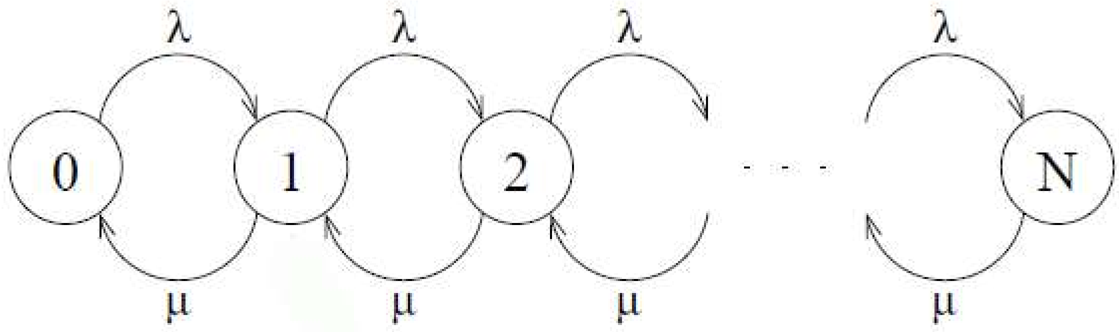
**The frequency of the events that so long time (measured in seconds) elapsed between the arrival of two consecutive packets*

Problems with Measurements

- The examined system MUST exist
  - It is costly and time consuming to build it
  - Its building elements may not be available yet
- A measurement is an intervention into the operation of the examined system
  - Is it allowed? Trust, legal, security, safety problems
- Problems and cost of the execution
  - Measurement devices, experts, …
  - Collection and evaluation of the results

Analytical Method

- We have good models for simple cases
  - Queueing theory is well established
    - Markov chains



    - Poisson process: the new requests arrive with exponential inter-arrival time
  - Simple cases do not characterize our real systems well enough
  - There are solvability problems in complex cases

Simulation: Model Creation

- We build a *computer executable model* of the system to be examined
  - The model is a simplified version of the system to be examined
  - The model can be executed by computer
  - The model characterizes the system well regarding the properties important for us
    - Medicine tests: a mouse is a model of humans
    - At a shop window: manikin is a model of humans
  - We always build a model <u>for a purpose</u>!

Simulation: Experimenting

- We design the experiments (depending on the aims of our investigations)
- We perform the experiments on the model and collect “measurement” results
- We evaluate our results
  - We may get answers to our questions (ready)
  - We may need further experiments
    - → go back to the design of experiments
  - We may need to improve our model
    - → go back to model building

Advantages of Simulation

- Can be carried out if the system to be examined (or even its building elements!) do not exist yet
- Types of experiments may be carried out, which are impossible on a real system
  - e.g. nuclear power plant, Internet
- May have much less cost than building a real system or even than experimenting on an existing system
- Good models may produce sufficiently accurate results

Limitations of Simulation

- Model building is time consuming
- Detailed models may result in unacceptably long execution time
- How well the results of the simulation characterize the modelled system?
  → Validation

Simulation of ICT Systems – Step 1

- Model creation
  - Model the building elements of the system
    - Study the operation of the given building element
    - Model its properties important for our purposes
  - Model the traffic of the System
    - Either based on traffic measurements on the modelled system
    - Or by modelling the applications
  - Build the model of the system from the elements
    - Need to acquire the topology of the system, the exact types of the elements and their configuration data

Detour: Stakeholders of Model Building

- We suppose that a consultation company works for a company operating a large system
  - The one who knows the real system (e.g. its planner, builder or operator)
  - Simulation professional (knows the methodology of the simulation and the used modelling system)
  - The management of the companies involved
- Important: What can each stakeholder expect from the project?

Simulation of ICT Systems – Step 2

- Design of the experiments – depends on the purpose of the analysis, it can be
  - To find the bottleneck(s)
  - To find the root cause of anomalies
  - Preparations for the introduction of new services
  - Preparations for reconstruction, update, adding further elements, etc.
- Important
  - What input parameters to be used?
  - What should be observed?

Simulation of ICT Systems – Step 3

- Execution of the experiments
  - Collect “measurement” data
  - Continue/repeat the simulation, until the required statistical accuracy of the measurement data is achieved

Simulation of ICT Systems – Step 4

- Evaluation of the results
  - Machine processing of a high amount of data
  - Filter out events interesting for us
    - E.g. the utilization of a resource is > 90%
  - Presentation of the results
    - Using tables, graphs, perhaps on the network topology
- Draw conclusions
  - Refine the model, design/execute new experiments, if necessary
- Presentation to the customer (management)

Types of simulation

- According to the time of the model
  - Continuous: the state of the system changes continuously
    - E.g.: water flows in a pipeline
  - Discrete: the state of the system changes in well defined discrete time points or we take them into consideration in this way
    - E.g. digital circuit, computer network, telephone network

    Called: Discrete-Event Simulation (DES)

Types of Discrete-Event Simulation

- According to the operation of the algorithm
  - Time-driven: the time of the model is increased by fixed steps
    - E.g.: in digital telephony, sampling happens in every 125µs
  - Event-driven: the occurrence of the events drives the operation of the simulation
    - This is more general, we deal with this one

Operation of the event-driven DES

- Future Event Set (FES)
  - Stores the events to occur in the future

Timestamp	Event
0.000000	Transmission of frame #0 begins
0.000100	The head of frame #0 arrives to the receiver
0.000512	Transmission of frame #0 ends
0.000612	The tail of frame #0 arrives to the receiver

Algorithm of the event-driven DES

```
Initialization: put staring event(s)
 into the FES;
REPEAT
  Remove the event with the lowest
   timestamp from the FES;
  NOW := the timestamp of the removed
   event;
  Process the event and schedule new
   events, if necessary;
UNTIL (NOW > limit)OR(no more events)OR
 (we must stop for some reason)
```

Basic concepts of event-driven DES

- Virtual time (also called: model time)
  - The time measured in the model
- Execution time (also called: wall clock time)
  - The time measured by the real-time clock of the computer executing the simulation

  - Causality requires that virtual time as a function of execution time be non-decreasing.
    - It is allowed that multiple events have the same timestamp. Execution of events with equal timestamps happens in the order they were put into the FES (unless overridden by their priorities).

A few definitions to learn

- **Modelling** (or model creation) is a *human activity* that builds a kind of (typically simplified) variation of a real (existing or imagined) system, which can be handled by a toolset (called modelling system).
- **Simulation** is an *experiment* carried out using a computer executable model.

A few definitions to learn

- **Emulation** is the *replacement* of a software or hardware by another software or hardware, the operation of which as a black box is the same as that of the original one, but its internal operation may be totally different.
- Comparison:
  - Simulation: experimenting
    - A flight simulator can be used for pilot training, but we can not travel by it
  - Emulation: replacement for normal use
    - An emulated CPU must give the same result as the original

A few definitions to learn

- **Monte Carlo simulation** is a special case of simulation; it follows random events and neglects some precise timing.
- **Trace-driven simulation** uses the precise data set of events experienced in a real system as its input.

A few definitions to learn

- **Verification** checks if the model is well implemented in the simulator. (debugging)
  Question: “Does this program work correctly?”
- **Validation** checks if the given model represents the real system well, and if questions regarding the real system can be answered using the given model.
  Question: “Is this program the right one?”

How to Handle Measurement Results?

- Measurement results may come from
  - Measurements taken on a real system
  - Measurements taken during simulation
- The rule is the same
  - **Measurements MUST be executed multiple times!**
  - The necessary number of repetitions to achieve reliable results depends on the actual measurement.

How to Handle Multiple Results?

- We use a summarizing function, usually either *average* or *median*.
  - Average is more inclusive and less sensitive to noise, but it is more sensitive to outliers.
- Using only a single number would result in oversimplification.
- To express indices of dispersion we can use
  - Standard deviation
  - Minimum and maximum
  - Percentiles, e.g. 1st percentile and 99th percentile

A Simple Example

minimum -------->	**1370**	1371	1100
10 percentile --->	**1390**	1389	1390
	1410	1411	1410
	1430	1429	1430
	1450	1451	1450
median ---------->	**1500**	1499	1500
	1550	1551	1550
	1570	1569	1570
	1590	1591	1590
90 percentile --->	**1610**	1609	1610
maximum ------->	**1630**	1631	1630
average:	1500	1500.09	1475.455
standard deviation:	94.34	94.35	150.16

Results of Measurement Series…

- In practice, we often use several input parameters in simulation. For example:
  - Number of clients
  - Number CPU cores used in the router
- They can have a few typical meaningful values
  - It is called “parameter study”, when all their possible combinations are used. Their results form…
    - A (single dimensional) table (1 parameter)
    - A matrix (2 parameters)
    - An N dimensional hypercube (N parameters)

Tips for Presentation of Results

- Graphs are easier to overview than tables, but graphs do not always substitute tables.
- Try to add all possible information to the figures.
- At the same time, require minimum effort from the readers to interpret the results.
- The validity of the results should also appear (in the Figure, or in its caption.)

Example 1

- Is there an easy to recognize trend?
- How many graphs are condensed into a single one?
- What could error bars mean?

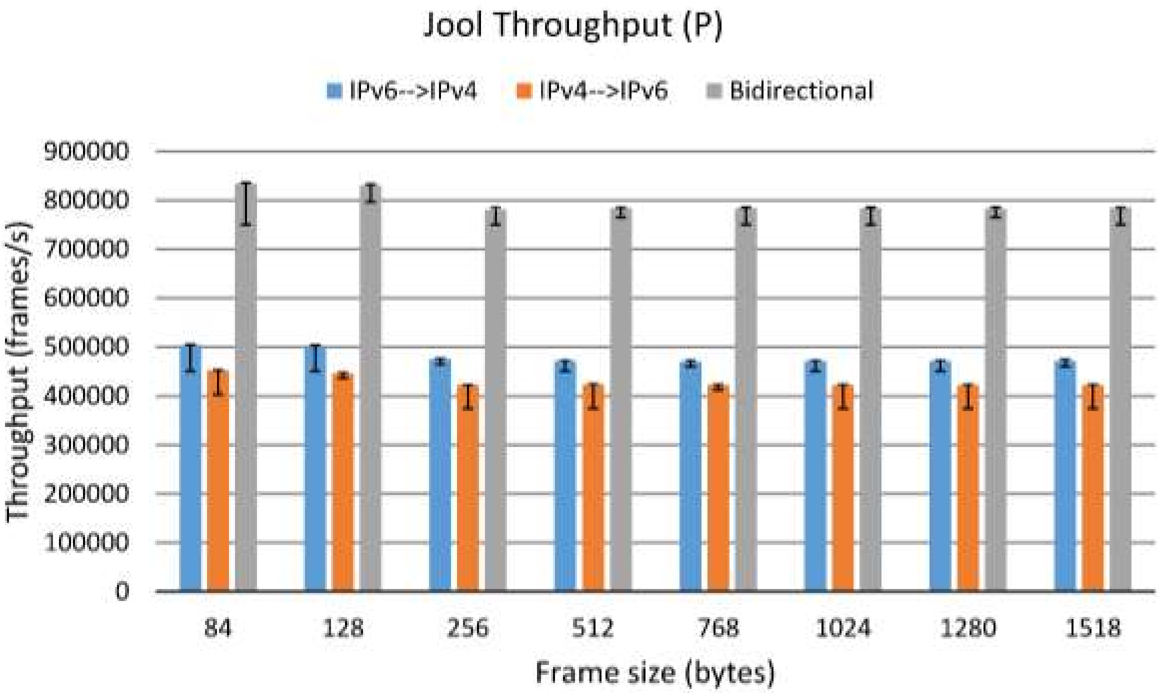


Fig. 4. Throughput results of Jool, TS2.

Source:

G. Lencse, K. Shima, "Performance Analysis of SIIT Implementations: Testing and Improving the Methodology“, *Computer Communications*, vol. 156, no. 1, pp. 54-67, April 15, 2020, DOI: 10.1016/j.comcom.2020.03.034

Example 2

- Is there an easy to recognize trend?
- How many input parameters can you observe?
- What other tip is demonstrated?

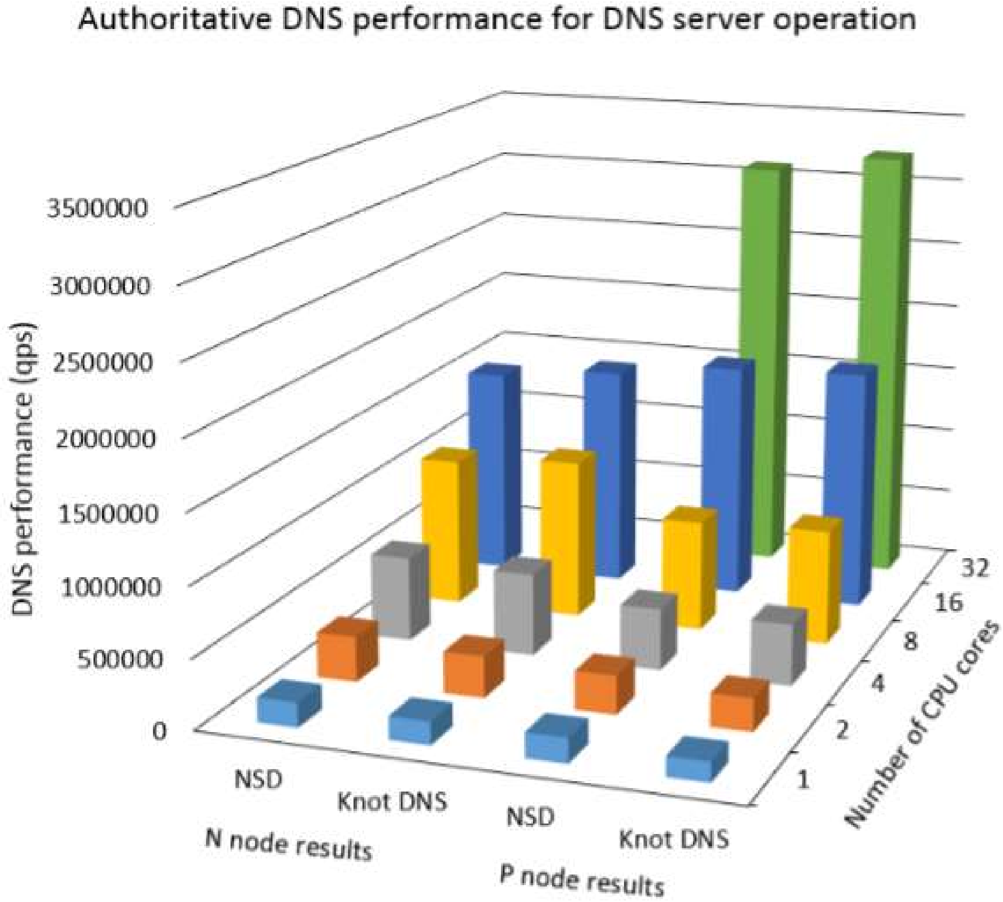


FIGURE 3. Comparison of NSD and Knot DNS for DNS server operation. (The N node result of Knot DNS at 16 cores are limited by Tester performance.)

Source:

G. Lencse, "Benchmarking Authoritative DNS Servers“, *IEEE Access*, vol. 8. pp. 130224-130238, July 2020. DOI: 10.1109/ACCESS.2020.3009141

Example 3

- Can you see a valid reason for using a table (and not a graph)?

Table 13 DNS64 Performance: BIND, Forwarder, Opteron

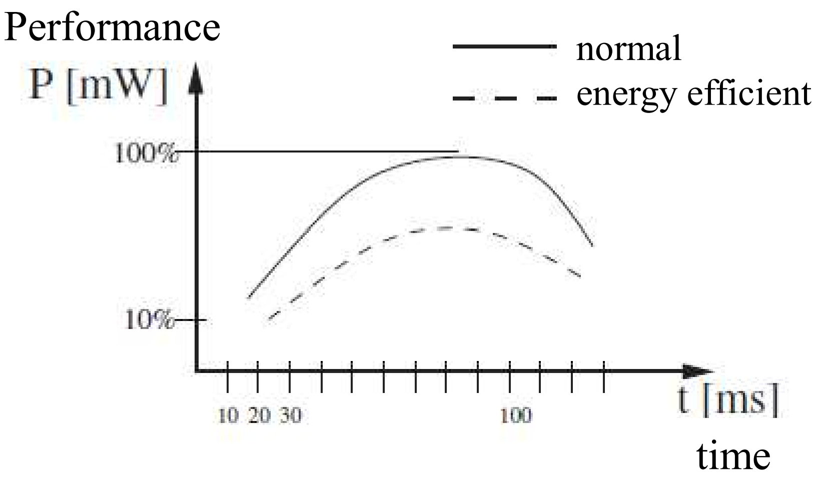
1	**Operating System**		**Linux**				**OpenBSD**				**FreeBSD**			
2	**Number of clients**		**1**	**2**	**4**	**8**	**1**	**2**	**4**	**8**	**1**	**2**	**4**	**8**
3	**Exec. time of**	average	0.067	0.098	0.213	0.409	0.094	0.188	0.382	0.783	0.082	0.121	0.239	0.479
4	**256 DNS**	std. dev.	0.005	0.007	0.017	0.018	0.006	0.005	0.007	0.015	0.010	0.005	0.008	0.012
5	**queries (s)**	maximum	0.140	0.170	0.290	0.530	0.180	0.260	0.440	0.860	0.490	0.160	0.290	0.540
6	**CPU utiliza-**	average	57.91	72.20	63.00	69.92	26.15	27.22	27.42	27.28	66.88	87.82	88.68	89.52
7	**tion (%)**	std. dev.	0.95	1.38	3.02	2.45	0.85	0.87	0.98	0.93	2.10	1.76	1.76	1.74
8	**Memory cons. (MB)**		49.992	80.691	147.262	277.242	38.324	67.652	123.117	233.848	62.609	94.406	169.414	303.613
9	**Number of requests served in a second (req./s)**		3838	5208	4816	5003	2721	2724	2682	2615	3130	4215	4290	4272

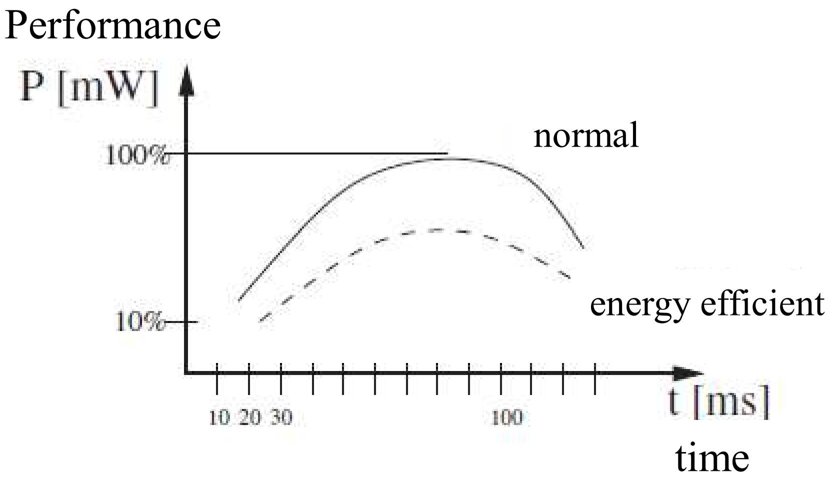
Source:

G. Lencse and S. Répás, "Performance Analysis and Comparison of Four DNS64 Implementations under Different Free Operating Systems“, *Telecommunication Systems* (Springer), vol 63, no 4, pp. 557-577, DOI: 10.1007/s11235-016-0142-x

Tips for Presentation of Results

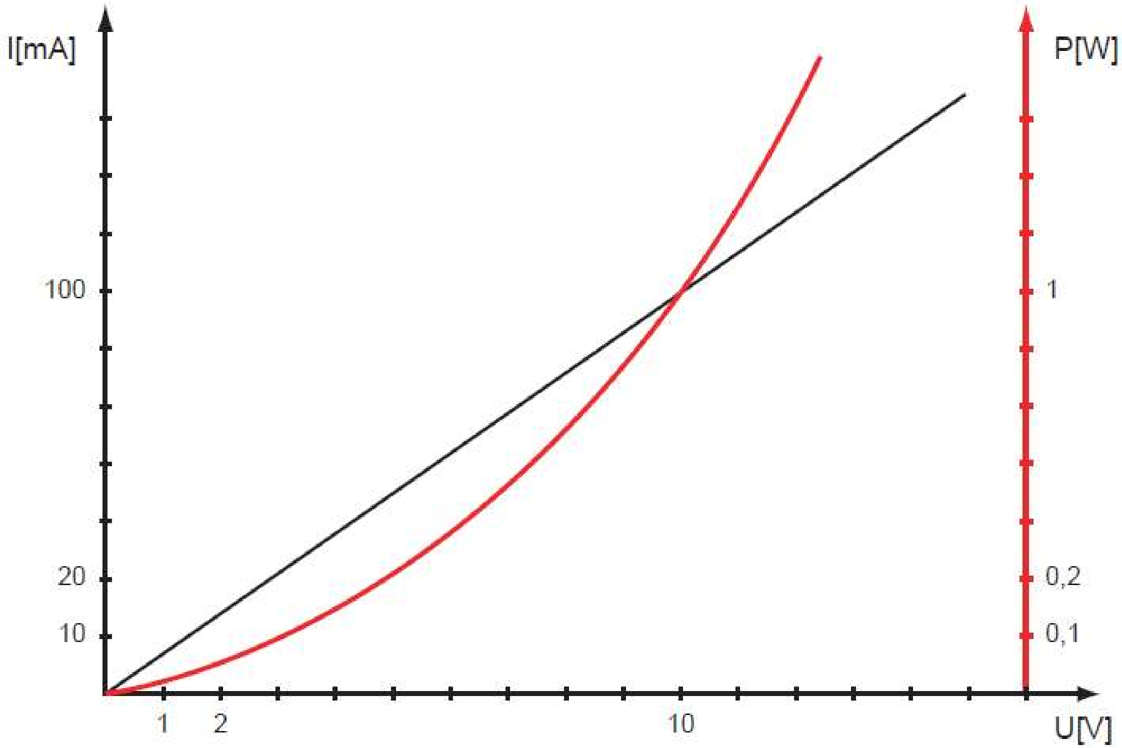
- Use a legend, or if possible, write in the meaning of the graphs into the figure





Tips for Presentation of Results

- Multiple quantities may be displayed as a function of the same independent variable (on the horizontal axis), but no more than two scales should appear on the vertical axis.



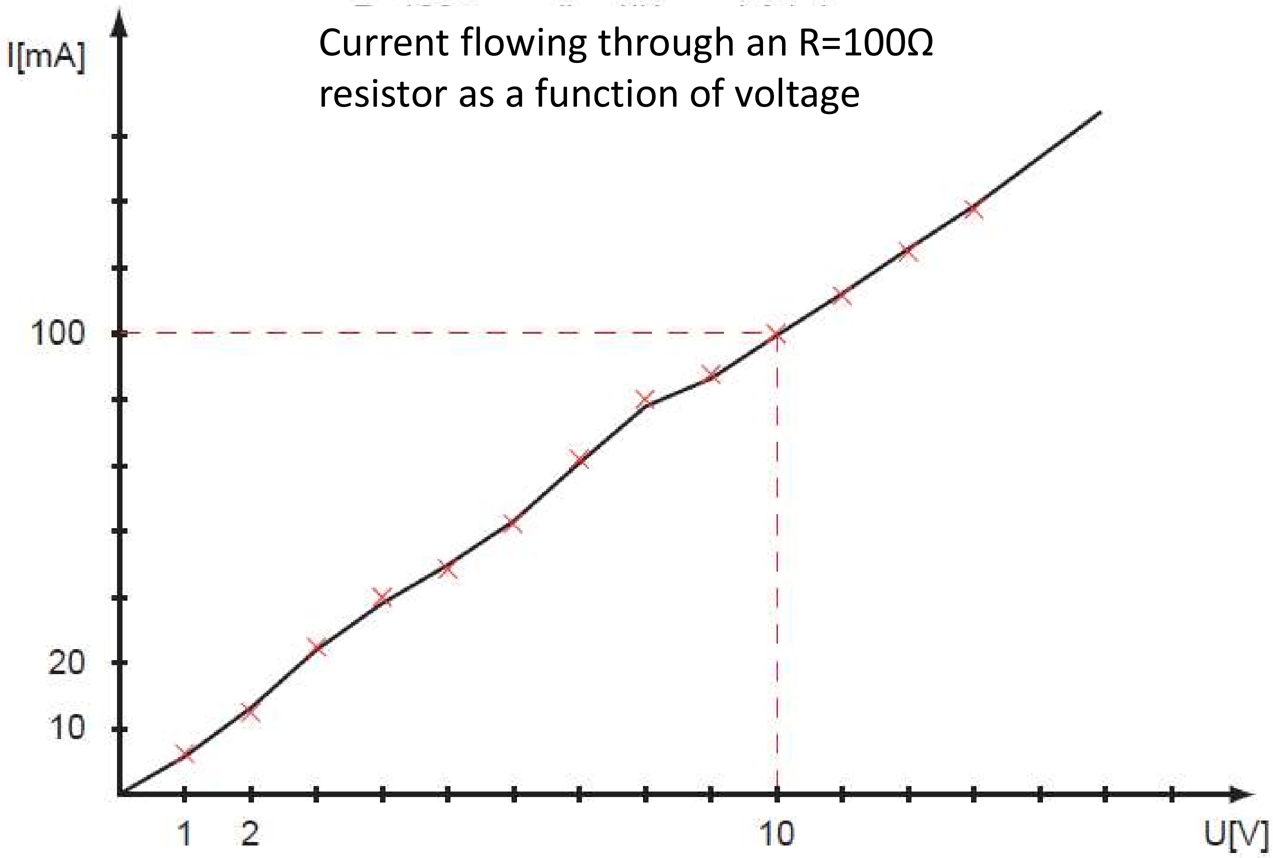
Tips for Presentation of Results

- It is worth using colors, if possible
- Choice of colors
  - Dark on white or vice versa is good

    But this is hardly readable on a projector
  - Select colors that look good together
    - Let us prepare beautiful presentations and papers. ☺
  - Our work should remain recognizable in black and write printing, too
    - We can use different line widths, different line styles (e.g. dashed, dotted lines, etc.)

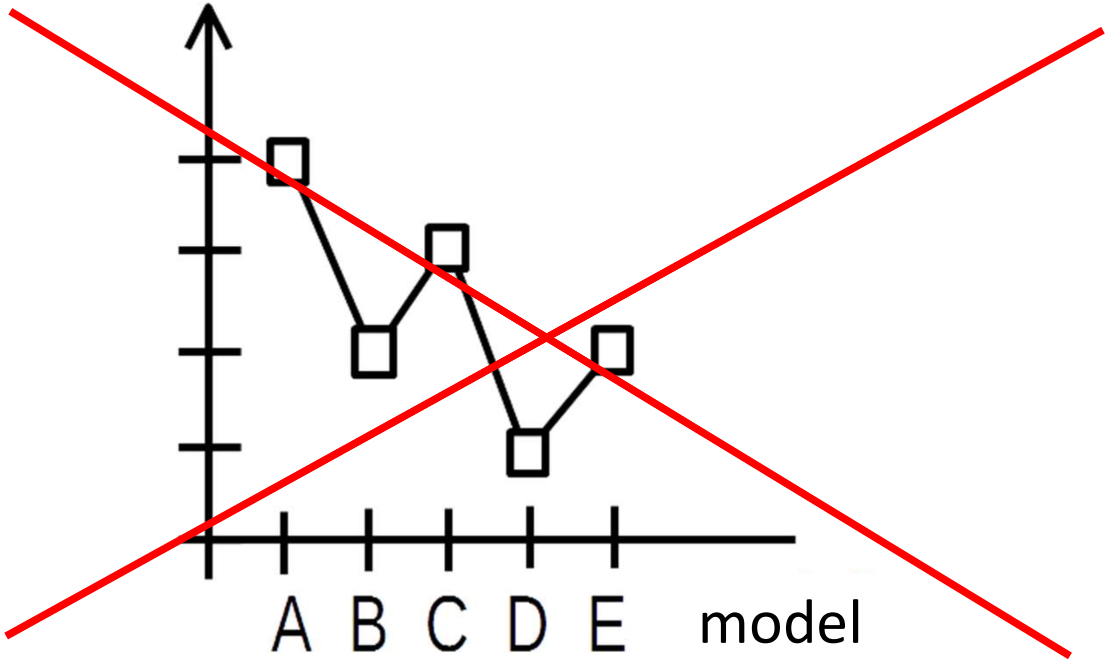
Tips for Presentation of Results

- If measurement results belong to a continuous function, they can be connected



Tips for Presentation of Results

- If measurement results do not belong to a continuous function, they should not be connected!

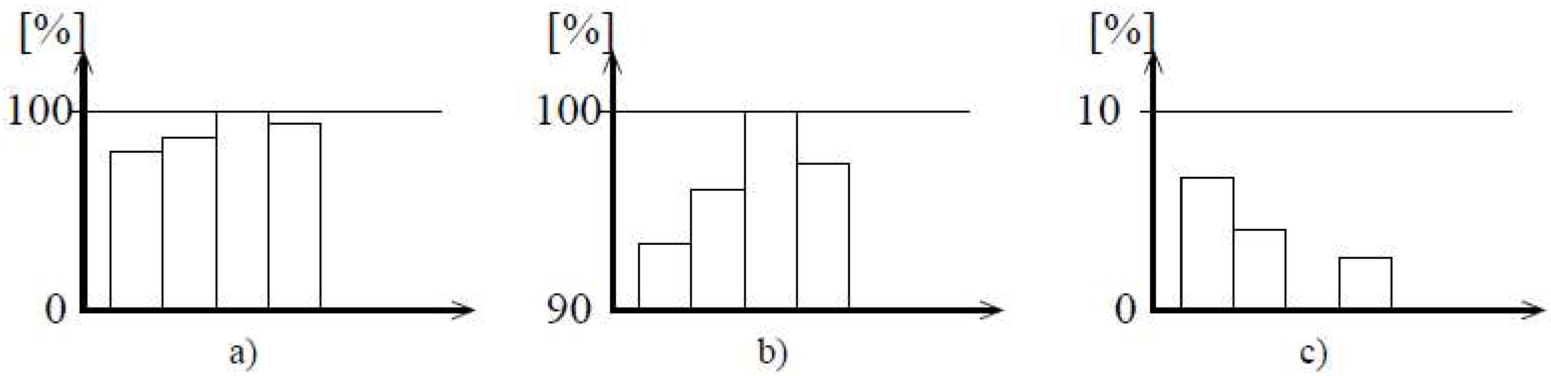


Tricks for Presentation of Results

- Tricks can be used for
  - emphasizing real facts, making things easier to observe
  - cheating
    - Please, do not do it!
    - But please be aware of it, especially when viewing ads!

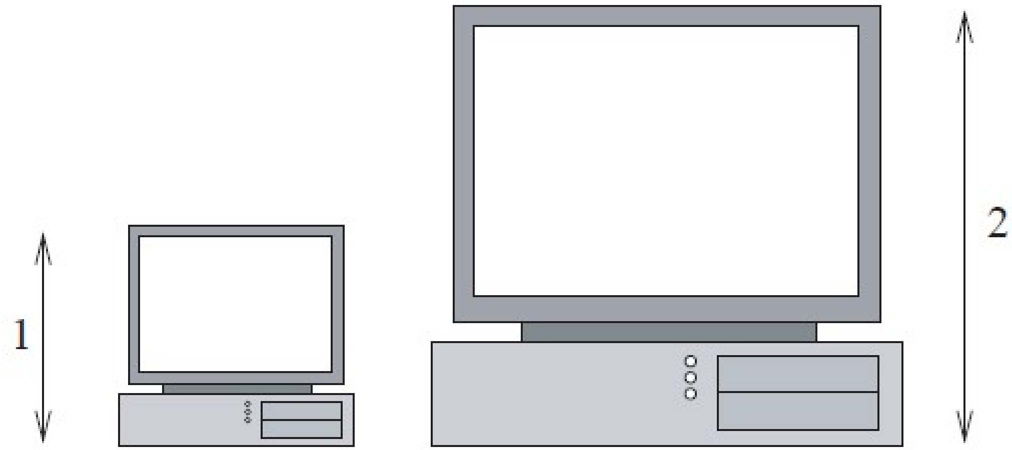
Tricks for Presentation of Results

- Gaming with the scale
  - We have exam results between 90% and 100%.
    a) expresses that all students are good
    b) shows the differences
    c) shows their mistakes



Tricks for Presentation of Results

- Instead of columns having equal widths, *two dimensional pictograms* are used
  - This is cheating: our vision senses their area: the twice as large icon seems to be four times as large!

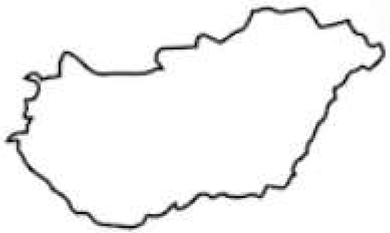


Tricks for Presentation of Results

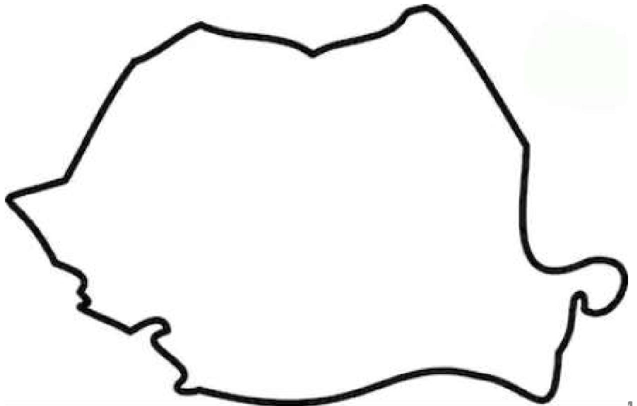
- Usage of certain *relative characteristics* is also suitable for <u>cheating</u>!
  - Number of wheels / Number of engine stokes
    - → An old East German car “Trabant” is better than a Lamborghini

4/2=2 > 4/4=1

  - Number of astronauts / number of inhabitants
    - → Hungary is better than Romania

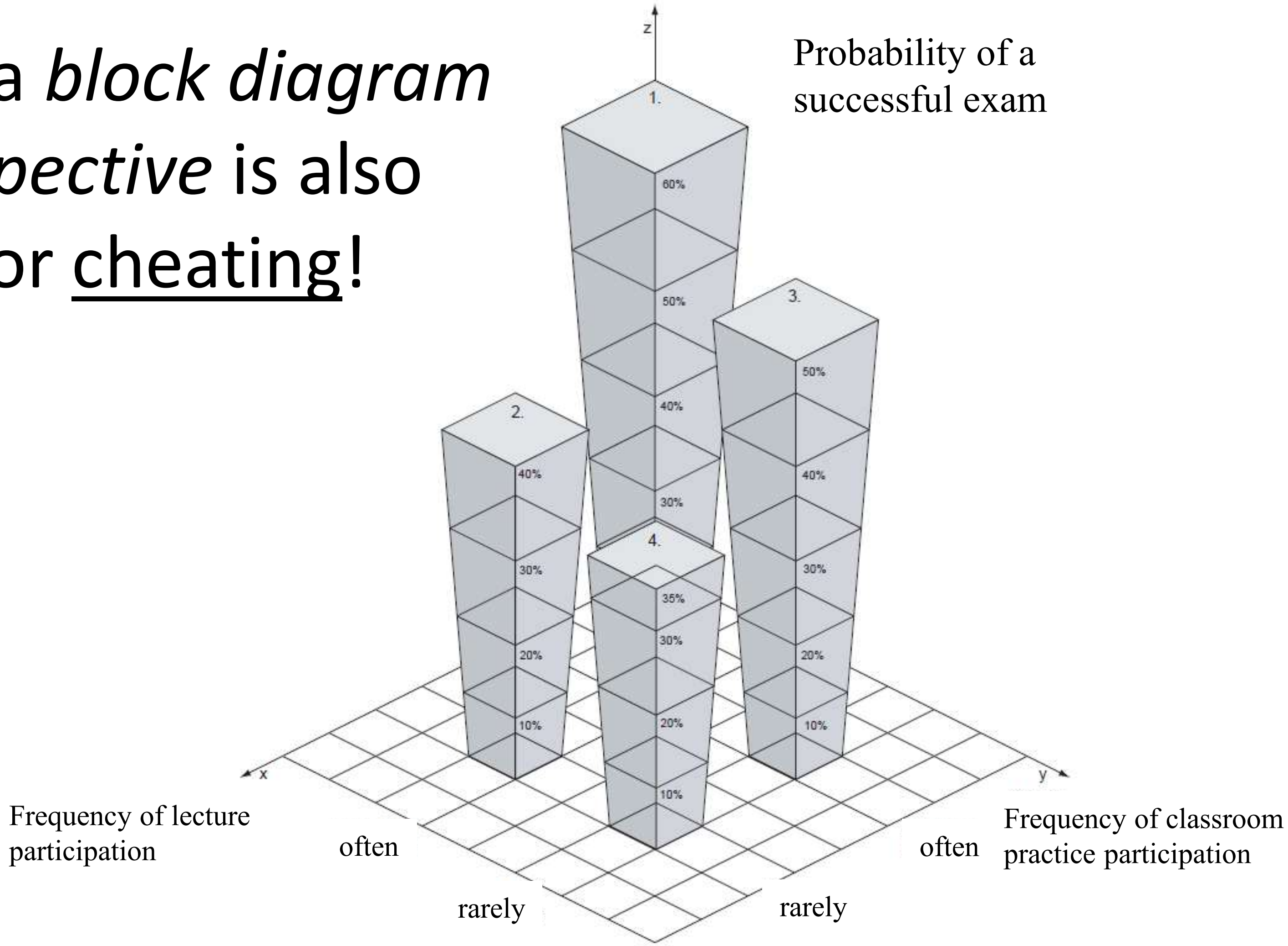


1/1,000,000 > 1/20,000,000



Tricks for Presentation of Results

- Usage of a *block diagram with perspective* is also suitable for cheating!



Sources

- Ray Jain, “The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling”, Wiley-Interscience, New York, 1991.
- György Pongor, “Kommunikációs rendszerek szimulációja”, an MSc course about simulation of ICT systems (in Hungarian), BME, VIK, 1993.

Thank you for your attention!

Questions?

Dr. Gábor Lencse
Professor
Dept. of Telecommunications, University of Győr
lencse@sze.hu