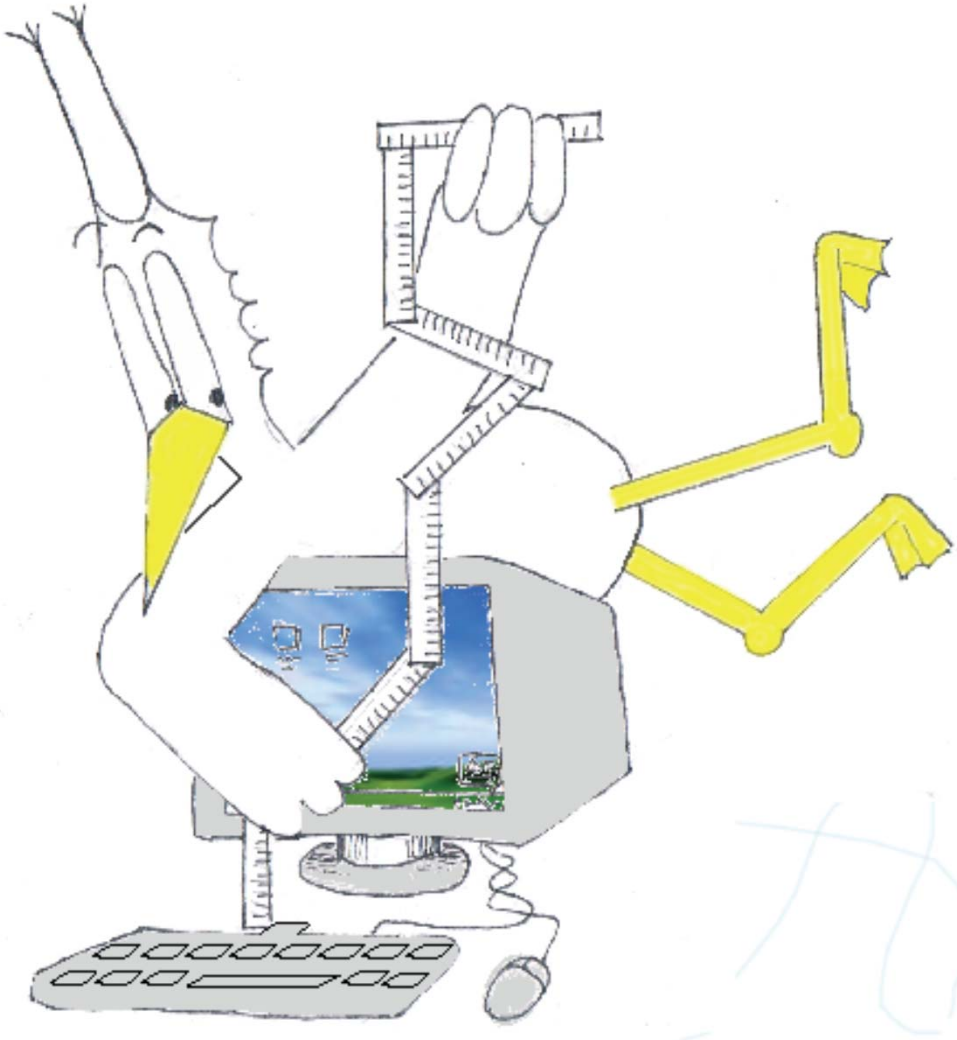
Methodology

easy but important



Jean-Yves Le Boudec, February 2019

Contents

Two Examples

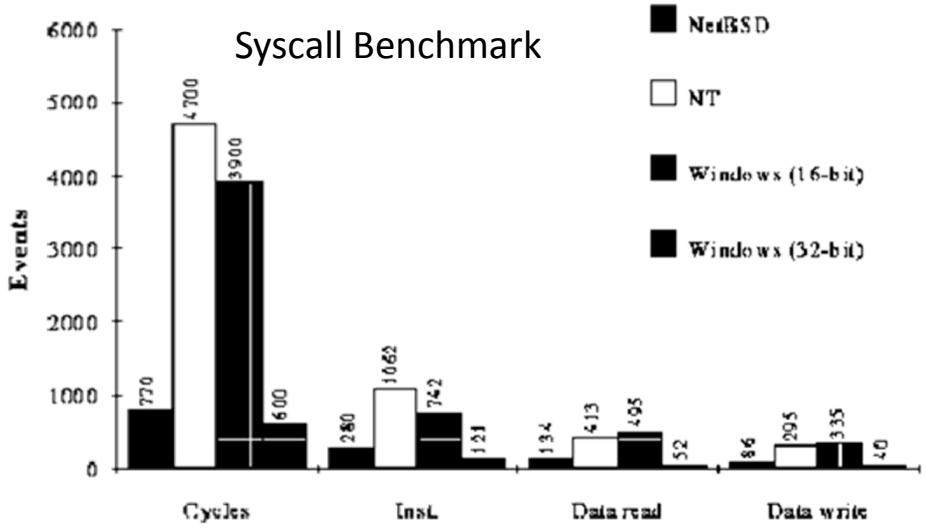
1. Compare Windows versus Linux
2. Evaluate the power consumption of a telecom «box»

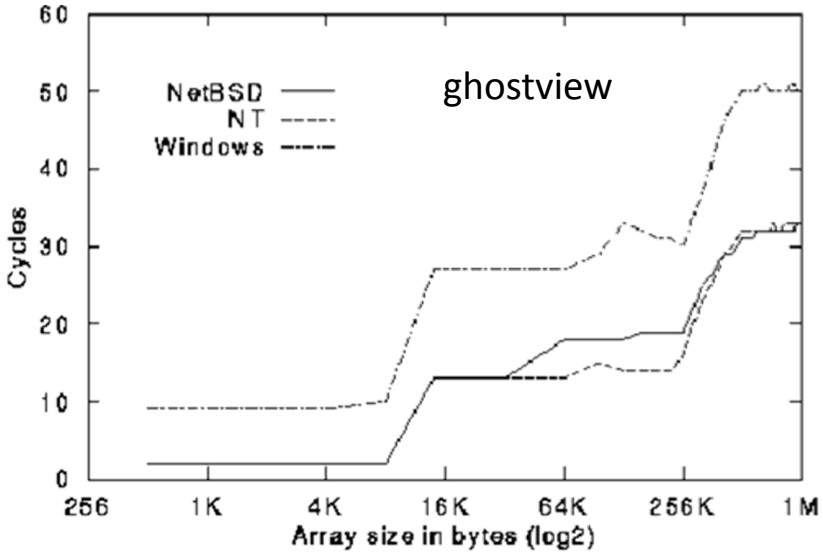
How would you tackle these problems ?

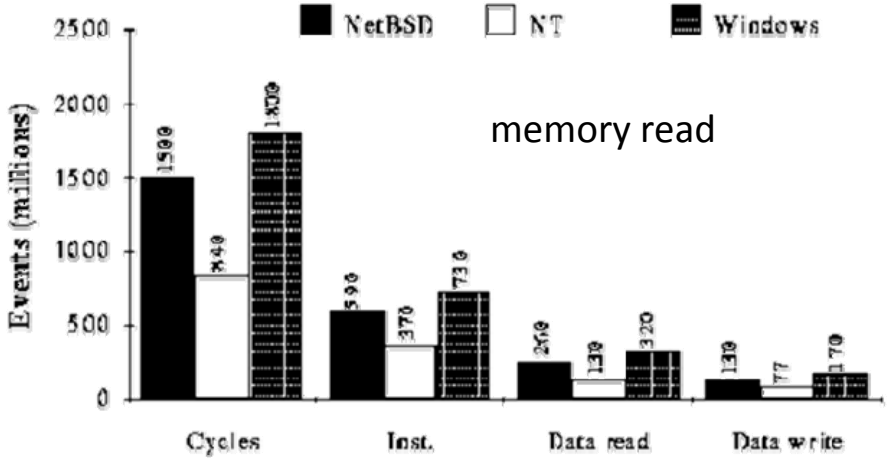
What do you need to be careful about ?

Example 1

EXAMPLE 2.5: WINDOWS VERSUS LINUX. Assume you want to compare Windows versus Linux. Chen and co-authors did it in [7]. They use as metric: number of cycles, instructions, data read/write operations. The load was generated by various benchmarks: "syscall" generates elementary operations (system calls); "memory read" generates references to an array; an application benchmark runs a popular application (ghostview).







Example 2

EXAMPLE 1.3: POWER CONSUMPTION. The electrical power consumed by a computer or telecom equipment depends on how efficiently the equipment can take advantage of low activity periods to save energy. One operator proposes the following metric as a measure of power consumption [2]:

$$P_{\text{Total}} = 0.35P_{\text{max}} + 0.4P_{50} + 0.25P_{\text{sleep}}$$

where P_{Total} is the power consumption when the equipment is running at full load, P_{50} when it is submitted to a load equal to 50% of its capacity and P_{sleep} when it is idle. The weights $(0.35, 0.4$ and $0.25)$ mean for example that we assume that the full load condition occurs during 35% of the time.

Metric

Define a metric; examples

- Response time
- Power consumption
- Throughput

Define operational conditions under which metric is measured (« Viewpoint », see Chapter 11)

EXAMPLE 2.5: WINDOWS VERSUS LINUX. Assume you want to compare Windows versus Linux. Chen and co-authors did it in [7]. They use as metric: number of cycles, instructions, data read/write operations. The load was generated by various benchmarks: "syscall" generates elementary operations (system calls); "memory read" generates references to an array; an application benchmark runs a popular application (ghostview).

Metric

EXAMPLE 1.3: POWER CONSUMPTION. The electrical power consumed by a computer or telecom equipment depends on how efficiently the equipment can take advantage of low activity periods to save energy. One operator proposes the following metric as a measure of power consumption [2]:

$$P_{\text{Total}} = 0.35P_{\text{max}} + 0.4P_{50} + 0.25P_{\text{sleep}}$$

where P_{Total} is the power consumption when the equipment is running at full load, P_{50} when it is submitted to a load equal to 50% of its capacity and P_{sleep} when it is idle. The weights $(0.35, 0.4$ and $0.25)$ mean for example that we assume that the full load condition occurs during 35% of the time.

EXAMPLE 1.3: MULTI-DIMENSIONAL METRIC AND KIVIAT DIAGRAM. We measure the performance of a web server submitted to the load of a standard workbench. We compare 5 different configurations, and obtain the results below.

Which configuration is best ?

Config	Power (W)	Response (ms)	Throughput (tps)
A	23.5	3.78	42.2
B	40.8	5.30	29.1
C	92.7	4.03	22.6
D	53.1	2.19	73.1
E	54.7	5.92	24.3

A. A
B. B
C. C
D. D
E. E
F. None of the above
G. I don't know

Load

You need to define the load under which your system operates

EXAMPLE 2.5: WINDOWS VERSUS LINUX. Assume you want to compare Windows versus Linux. Chen and co-authors did it in [7]. They use as metric: number of cycles, instructions, data read/write operations. The load was generated by various benchmarks: "syscall" generates elementary operations (system calls); "memory read" generates references to an array; an application benchmark runs a popular application (ghostview).

Load

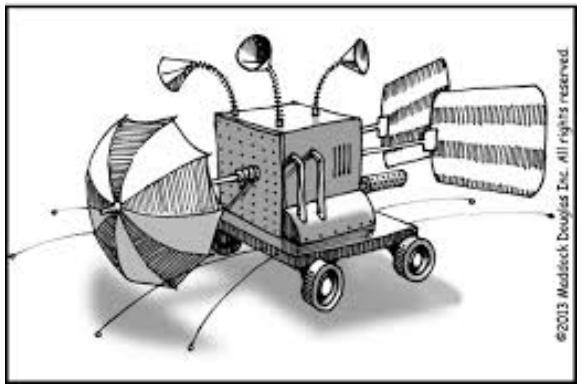
You need to define the load under which your system operates

EXAMPLE 1.3: POWER CONSUMPTION. The electrical power consumed by a computer or telecom equipment depends on how efficiently the equipment can take advantage of low activity periods to save energy. One operator proposes the following metric as a measure of power consumption [2]:

$$P_{\text{Total}} = 0.35P_{\text{max}} + 0.4P_{50} + 0.25P_{\text{sleep}}$$

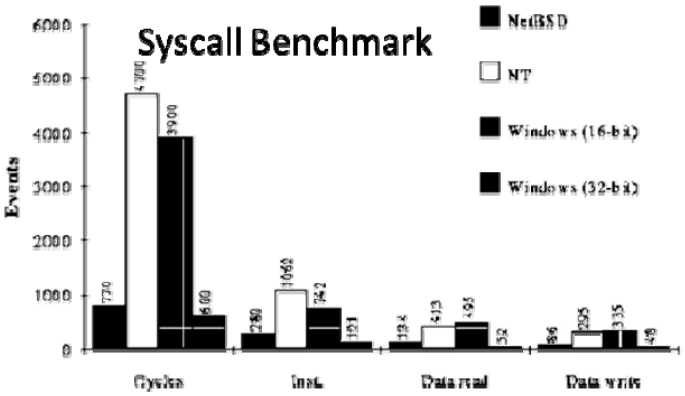
where P_{Total} is the power consumption when the equipment is running at full load, P_{50} when it is submitted to a load equal to 50% of its capacity and P_{sleep} when it is idle. The weights $(0.35, 0.4$ and $0.25)$ mean for example that we assume that the full load condition occurs during 35% of the time.

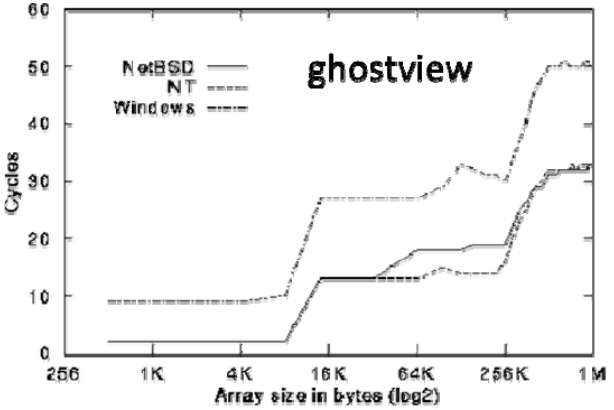
Load

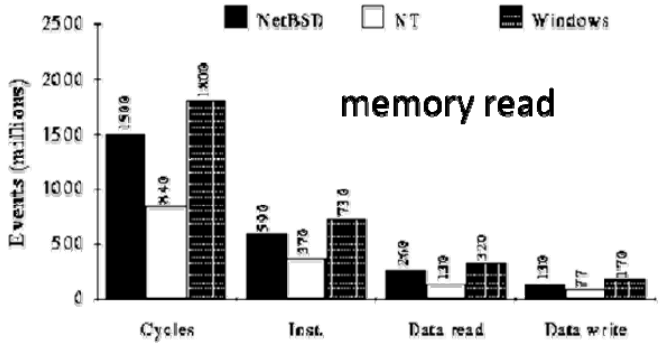


Metric

System under study







Where is the metric ?
Where is the load ?

Know your goals

EXAMPLE 2.5: WINDOWS VERSUS LINUX. Assume you want to compare Windows versus Linux. Chen and co-authors did it in [7]. They use as metric: number of cycles, instructions, data read/write operations. The load was generated by various benchmarks: "syscall" generates elementary operations (system calls); "memory read" generates references to an array; an application benchmark runs a popular application (ghostview).

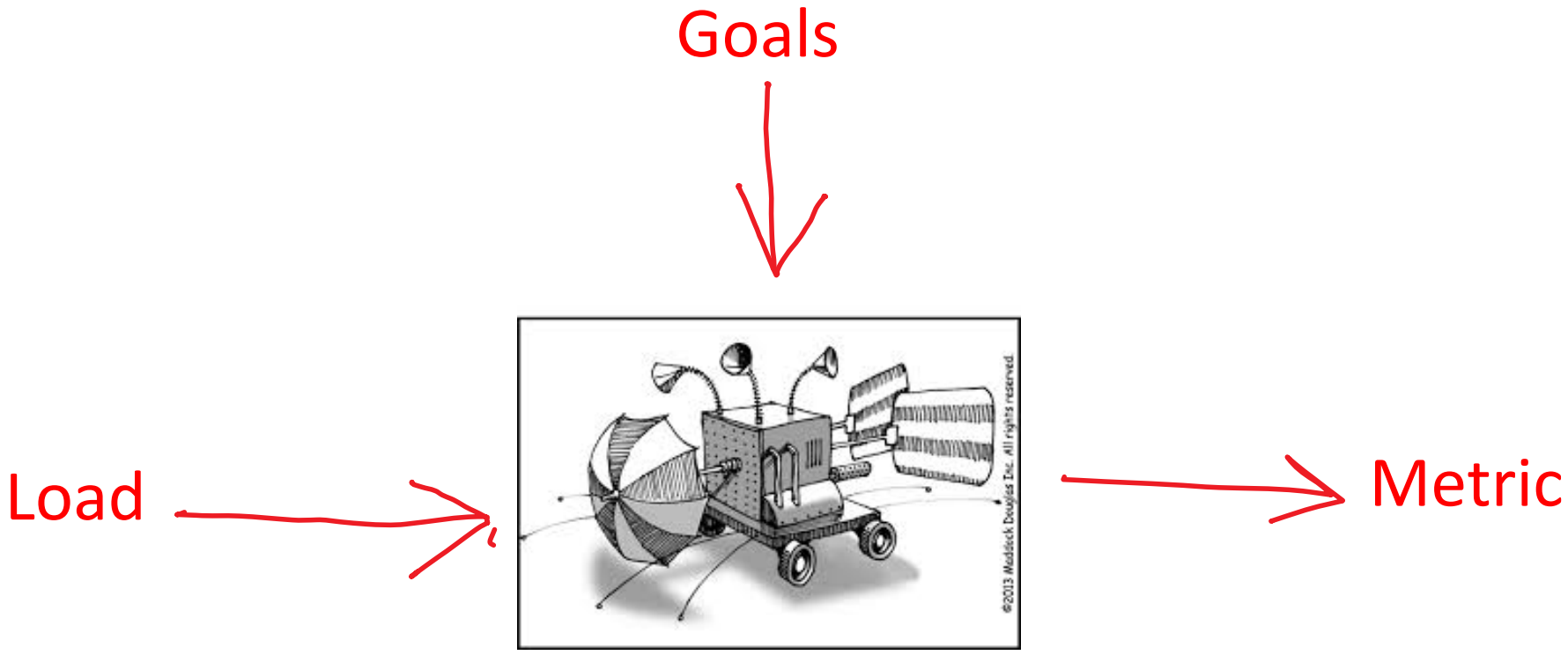
Goal in Example 2.5 is to make a *comparison*

EXAMPLE 1.3: POWER CONSUMPTION. The electrical power consumed by a computer or telecom equipment depends on how efficiently the equipment can take advantage of low activity periods to save energy. One operator proposes the following metric as a measure of power consumption [2]:

$$P_{\text{Total}} = 0.35P_{\text{max}} + 0.4P_{50} + 0.25P_{\text{sleep}}$$

Goal in Example 1.2 is to provide an *engineering rule*

Putting Things Together
A Performance Evaluation Study…

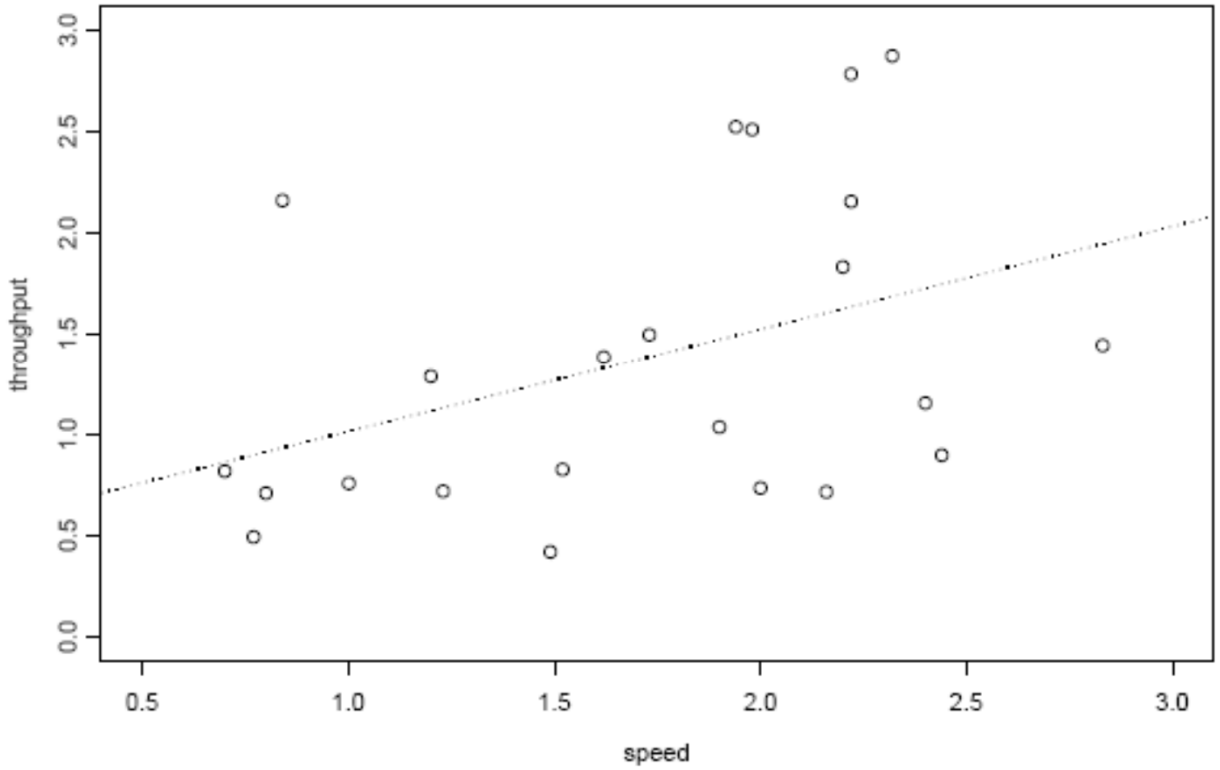


System under study

3. Factors

TCP Throughput Increases with Mobility

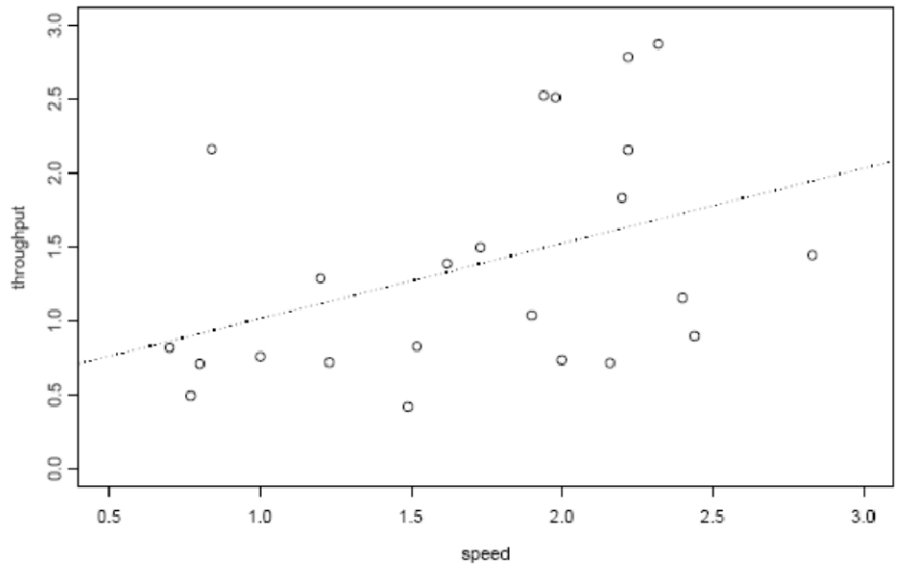
EXAMPLE 1.6: TCP THROUGHPUT. Figure 1.1, left, plots the throughput achieved by a mobile during a file transfer as a function of its speed. It suggests that throughput increases with mobility.



Does mobility increase throughput ?

A. Yes, it is proven by this experiment
B. It is true but perhaps only for a very specific system
C. No it is not true
D. I don't know

EXAMPLE 1.6: TCP THROUGHPUT. Figure 1.1, left, plots the throughput achieved by a mobile during a file transfer as a function of its speed. It suggests that throughput increases with mobility. The right plot shows the same data, but now the mobiles are



Simpson's Paradox

A well known phenomenon -- Special case of Hidden Factor paradox when metric is success rate and factors are discrete

We classify the mobiles as slow (speed ≤ 2m/s) or fast (speed > 2m/s). We obtain the following result. we say that a mobile is successful if its throughput is ≥ 1.5Mb/s

	failure	success		$\mathbb{P}$(success)
slow	11	3	14	0.214
fast	5	4	9	0.444
	16	7	23	

from where we conclude that fast mobiles have a higher success probability than slow

Now introduce the nuisance parameter "socket buffer size":

"S" mobiles	failure	success		$\mathbb{P}$(success)
slow	10	1	11	0.091
fast	1	0	1	0.00
	11	1	12	

"L" mobiles	failure	success		$\mathbb{P}$(success)
slow	1	2	3	0.667
fast	4	4	8	0.500
	5	6	11	

Berkeley Sex Case 1973 (source: wikipedia)

	Applicants	**% admitted**
Men	8442	**44%**
Women	4321	35%

However when examining the individual departments, it was found that no department was significantly biased against women; in fact, most departments had a small bias against men.

Major	**Men**		**Women**	
	Applicants	% admitted	Applicants	% admitted
A	825	62%	108	**82%**
B	560	63%	25	**68%**
C	325	**37%**	593	34%
D	417	33%	375	**35%**

Simpson's paradox explained

$$P(\text{good} \mid \text{slow}) = \sum_i P(\text{good} \mid \text{slow}, hf = i) P(hf = i \mid \text{slow})$$

$$P(\text{good} \mid \text{fast}) = \sum_i P(\text{good} \mid \text{fast}, hf = i) P(hf = i \mid \text{fast})$$

$\leq$ $\geq$

weights are different

good = high thruput
hf = hidden factor
hf ∈ {Large socket buffer, Small socket buffer}

"S" mobiles	failure	success		ℙ(success)
slow	10	1	11	0.091
fast	1	0	1	0.00
	11	1	12	

"L" mobiles	failure	success		ℙ(success)
slow	1	2	3	0.667
fast	4	4	8	0.500
	5	6	11	

$$P(hf = \text{Small socket buffer} \mid \text{slow}) = \frac{11}{14} \approx 79\%$$

$$P(hf = \text{Small socket buffer} \mid \text{fast}) = \frac{1}{9} \approx 11\%$$

Avoiding Simpson's Paradox

$$P(\text{good} \mid \text{slow}) = \sum_i P(\text{good} \mid \text{slow}, hf = i) P(hf = i \mid \text{slow})$$

$$P(\text{good} \mid \text{fast}) = \sum_i P(\text{good} \mid \text{fast}, hf = i) P(hf = i \mid \text{fast})$$

Make the weights equal !

$$P(hf = i|\text{slow}) = P(hf = i|\text{fast}), \qquad \forall i$$

$P(hf = i|\text{slow}) = P(hf = i|\text{fast}),\ \ \forall i$ means …

A. The hidden factor hf and the desired factor slow/fast are independent
B. The hidden factor hf is distributed uniformly across all experimental conditions
C. A and B
D. None of the above
E. I don't know

Take-Home Message

Identify hidden factors – make them disappear if you can

QUESTION 1.4.1. *Consider again comparing Windows versus Linux. Can you imagine what factors might play an important role in the analysis ? What external factors have to be taken care of during the evaluation ?* [2]

---

[2]From [1]: External factors are: background activity; multiple users; network activity. These were reduced to a minimum by shutting the network down and allowing one single user. The different ways of handling idle periods in Windows NT and NetBSD also need to be accounted for, because they affect the interpretation of measurements. Cycle counts in idle periods of NetBSD have to be removed.

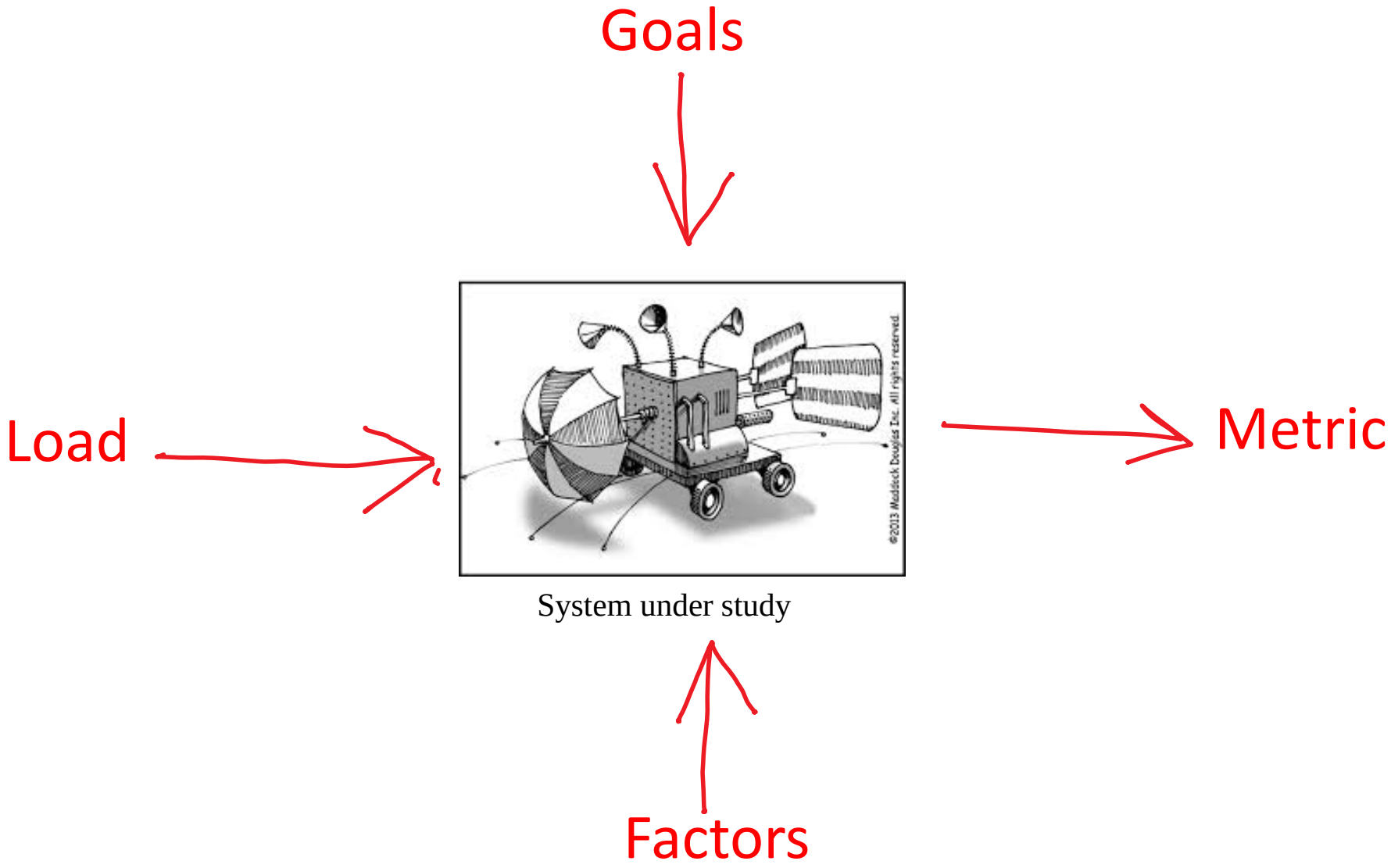
else make them appear explicitly in the analysis, or randomize the experiments to neutralize their impact

Take Home Message

Performance evaluation uses the language of probabilities

In this course we will exercise how to use probability theory in practice to do scientific work

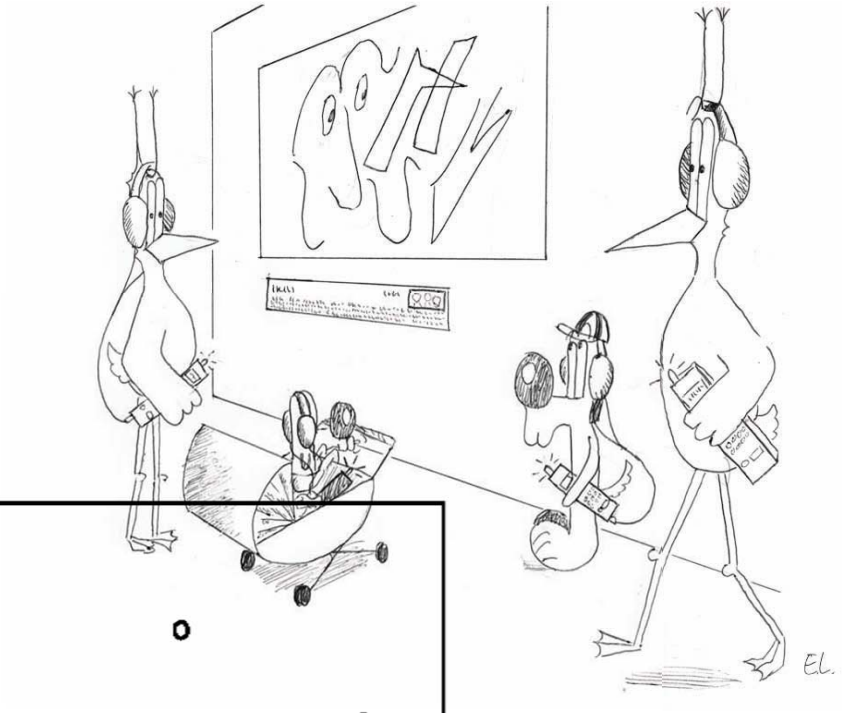
Putting Things Together
A Performance Evaluation Study…

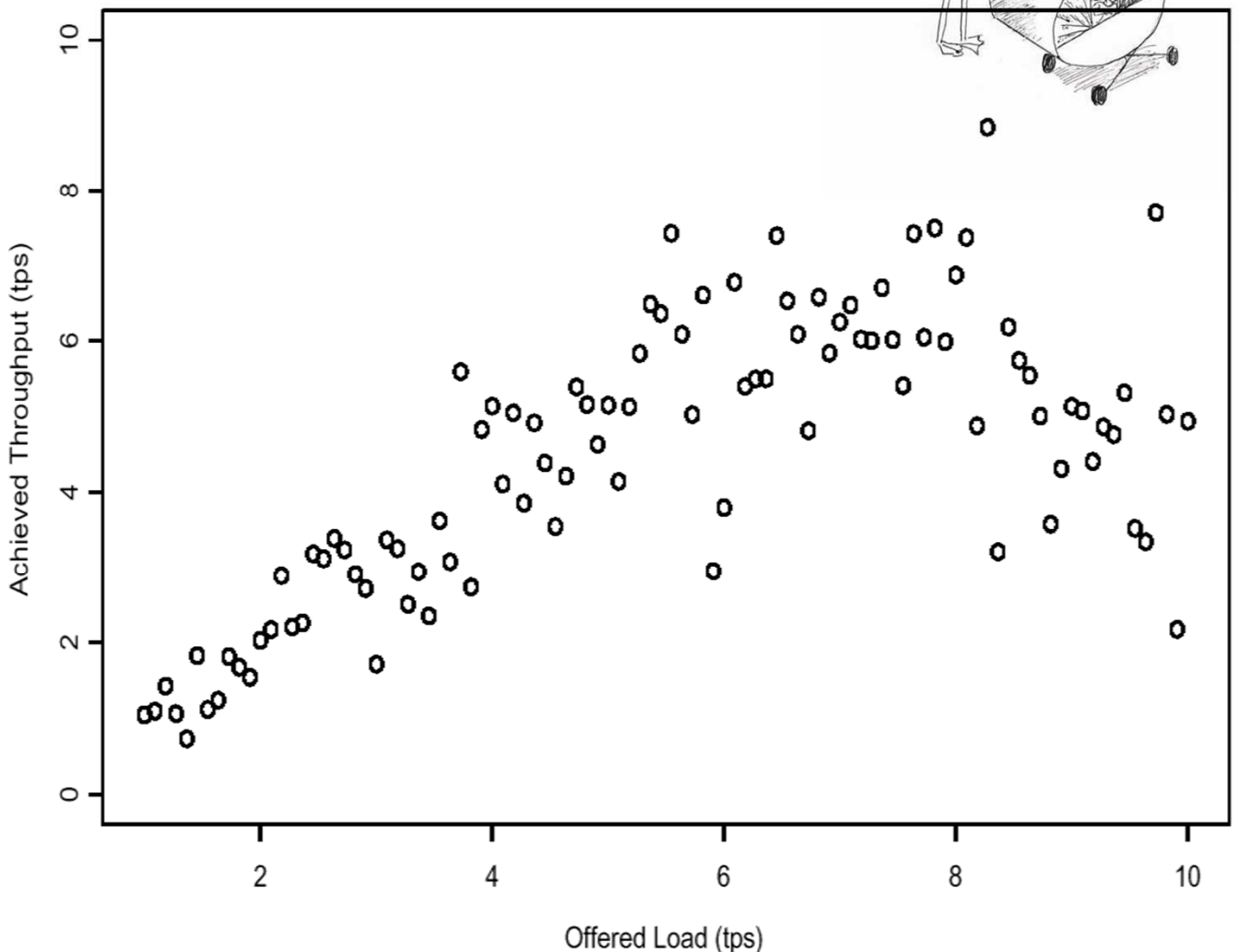


System under study

4. The scientific method

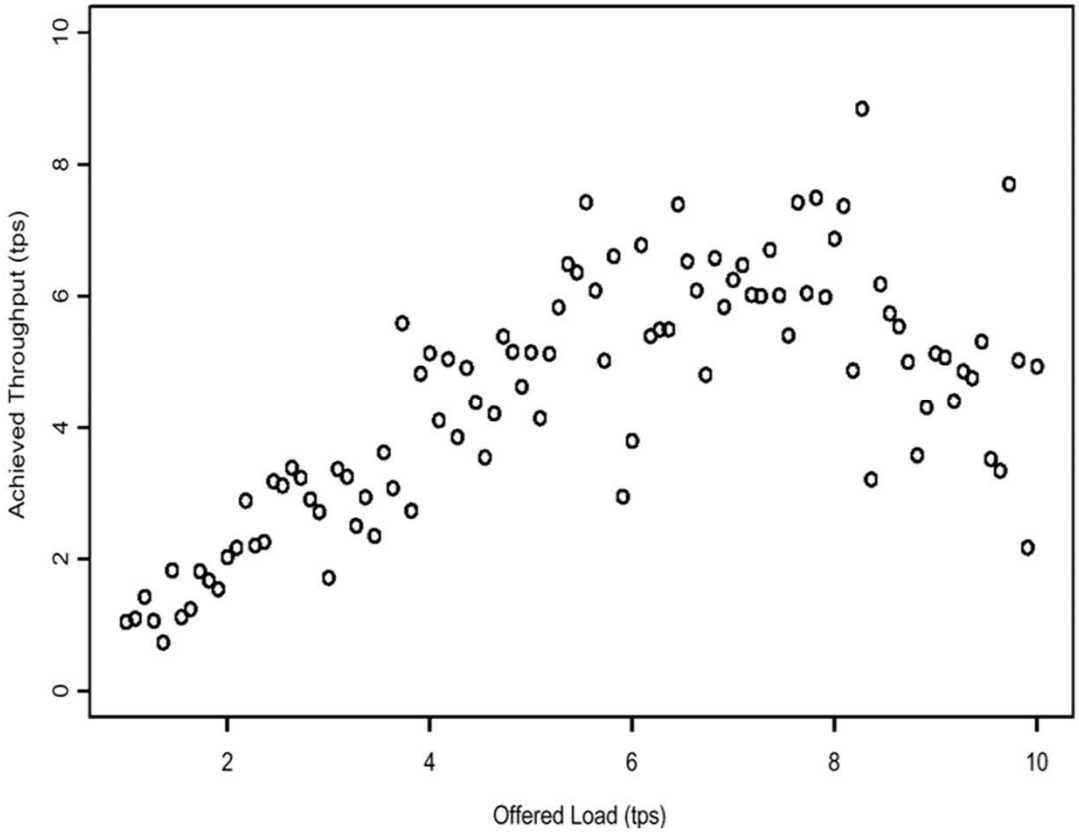
Joe measures performance of his Wireless Shop:





What do *you* recommend to Joe ?

A. Buy more access points
B. Change your server
C. Call IBM
D. Live with it
E. I don't know



Example 2:
Is ATM-ABR better than ATM-UBR ?

ABSTRACT. We compare the performance of ABR and UBR for providing high-speed network interconnection services for TCP traffic. We test the hypothesis that UBR with adequate buffering in the ATM switches results in better overall goodput for TCP traffic than explicit rate ABR for LAN interconnection. This is shown to be true in a wide selection of scenarios. Four phenomena that may lead to bad ABR performance are identified and we test whether each of these has a significant impact on TCP goodput. This reveals that the extra delay incurred in the ABR end-systems and the overhead of RM cells account for the difference in performance. We test

Take Home Message

You should not conclude from an experiment without trying to invalidate the conclusion

(Popper, 1934): you should alternate between the roles of

- Proponent
- Adversary

5. Patterns

These are common traits found in different situations

Knowing some of them may save *a lot of* time

EXAMPLE 1.11: BOTTLENECKS. You are asked to evaluate the performance of an information system. An application server can be compiled with two options, A and B. An experiments was done: ten test users (remote or local) measured the time to complete a complex transaction on four days. On day 1, option A is used; on day 2, option B is. The results are in the table below.

	remote	local
A	123	43
	189	38
	99	49
	167	37
	177	44

	remote	local
B	107	62
	179	69
	199	56
	103	47
	178	71

The expert concluded that the performance for remote users is independent of the choice of an information system. We can criticize this finding and instead do a bottleneck analysis. For remote users, the bottleneck is the network access; the compiler option has little impact. When the bottleneck is removed, i.e. for local users, option A is slightly better.

Bottlenecks may be your enemy

Bottlenecks are like non invited people at a party – they may impose their agenda

Previous example: what we are measuring is the bottleneck, not the intended factor

How do you proceed ?

EXAMPLE 1.12: CPU MODEL. A detailed screening of a transaction system shows that one transaction costs in average: 1'238'400 CPU instructions; 102.3 disk accesses and 4 packets sent on the network. The processor can handle 10^9 instructions per second; the disk can support 10^4 accesses per second; the network can support 10^4 packets per second. We would like to know how many transactions per second the system can support.

A. Do a queuing theory analysis
B. Do a simulation
C. None of the above
D. I don't know

Bottlenecks are Your Friends

Simplify your life, analyze bottlenecks !

In many cases, you may ignore the rest

Behind a Bottleneck May Hide Another Bottleneck

DOI:10.1145/1409360.1409

Want to make your Web site fly? Focus on frontend performance.

BY STEVE SOUDERS

High-Performa Web Site

Rule 6: Put Scripts at the Bottom

External scripts (typically, ".js" files) have a bigger impact on performance than other resources for two reasons. First, once a browser starts downloading a script it won't start any other parallel downloads. Second, the browser won't render any elements below a script until the script has finished downloading. Both of these impacts are felt when scripts are placed near the top of the page, such as in the HEAD section. Other resources in the page (such as images) are delayed from being downloaded and elements in the page that already exist (such as the HTML text in the document itself) aren't displayed until the earlier scripts are done. Moving scripts lower in the page avoids these problems.

Rule 7: Avoid CSS Expressions

CSS expressions are a way to set CSS

to serve JavaScript and CSS via external files, while making them cacheable with a far future Expires header as explained in Rule 3.

Rule 9: Reduce DNS Lookups

The Domain Name System (DNS) is like a phone book: it maps a hostname to an IP address. Hostnames are easier for humans to understand, but the IP address is what browsers need to establish a connection to the Web server. Every hostname that's used in a Web page must be resolved using DNS. These DNS lookups carry a cost; they can take 20–100 milliseconds each. Therefore, it's best to reduce the number of unique hostnames used in a Web page.

Rule 10: Minify JavaScript

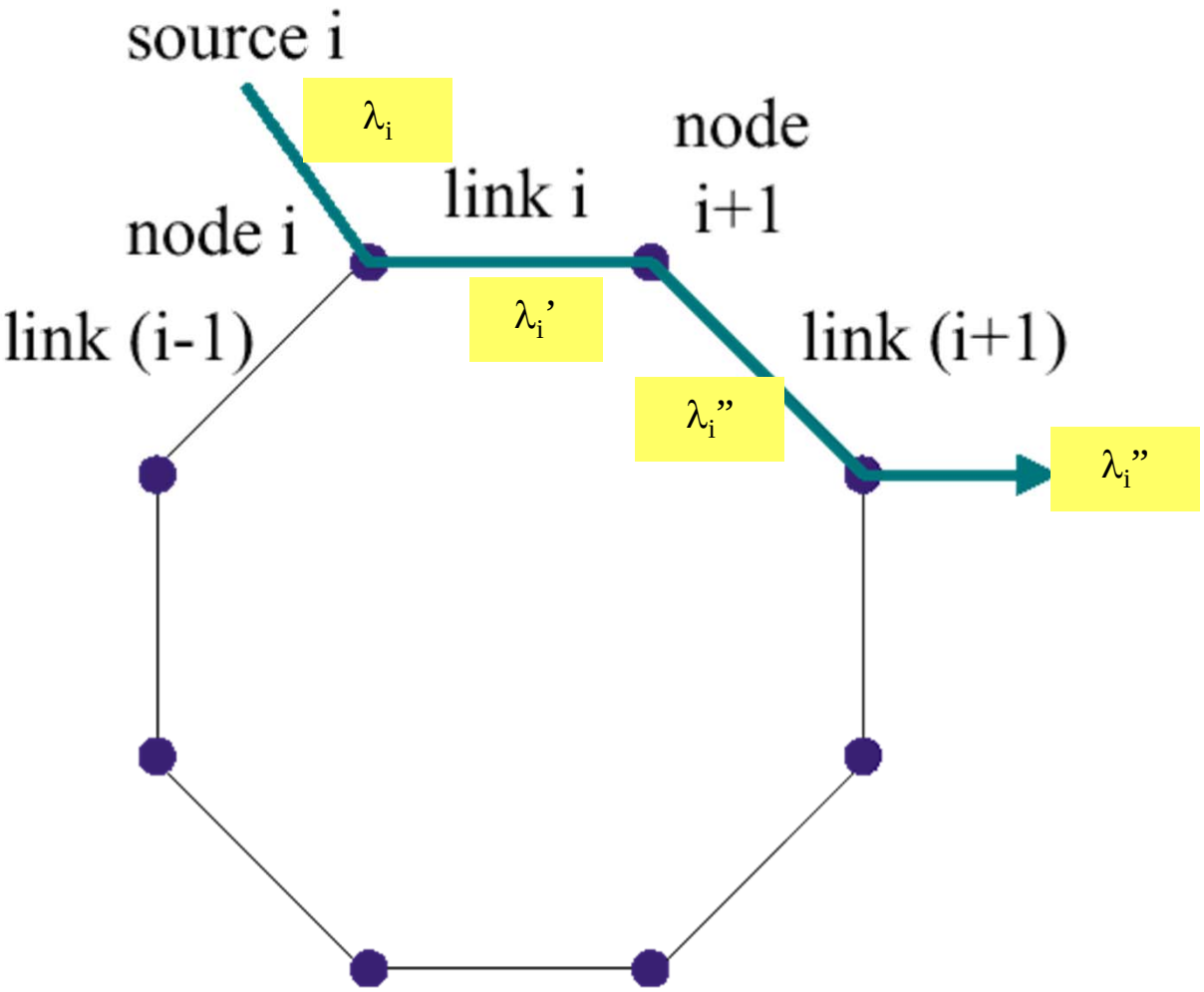
As described in Rule 4, compression is the best way to reduce the size of text files transferred over the Internet. The

would seem uncommon, but in a review of U.S. Web sites it could be found in two of the top 10 sites. Web sites that have a large number of scripts and a large number of developers are most likely to suffer from this problem.

Rule 13: Configure ETags

Entity tags (ETags) are a mechanism used by Web clients and servers to verify that a cached resource is valid. In other words, does the resource (image, script, stylesheet, among others) in the browser's cache match the one on the server? If so, rather than transmitting the entire file (again), the server simply returns a 304 Not Modified status telling the browser to use its locally cached copy. In HTTP/1.0, validity checks were based on a resource's Last-Modified date: if the date of the cached file matched the file on the server, then the validation succeeded. ETags were introduced in

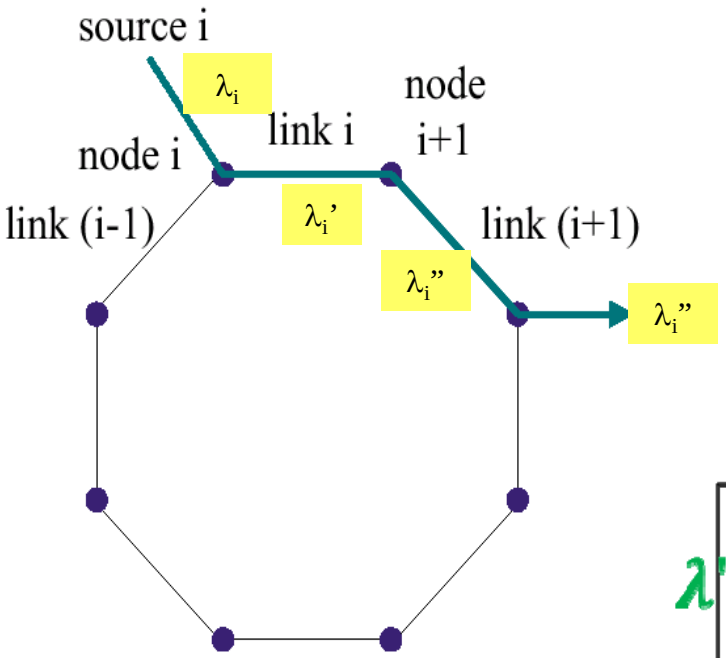
etc. The author describes 14 possible components, any of which, if present, is candidate for being the bottleneck, and suggests to remove all of them. Doing so leaves as bottlenecks network access and server CPU speed.

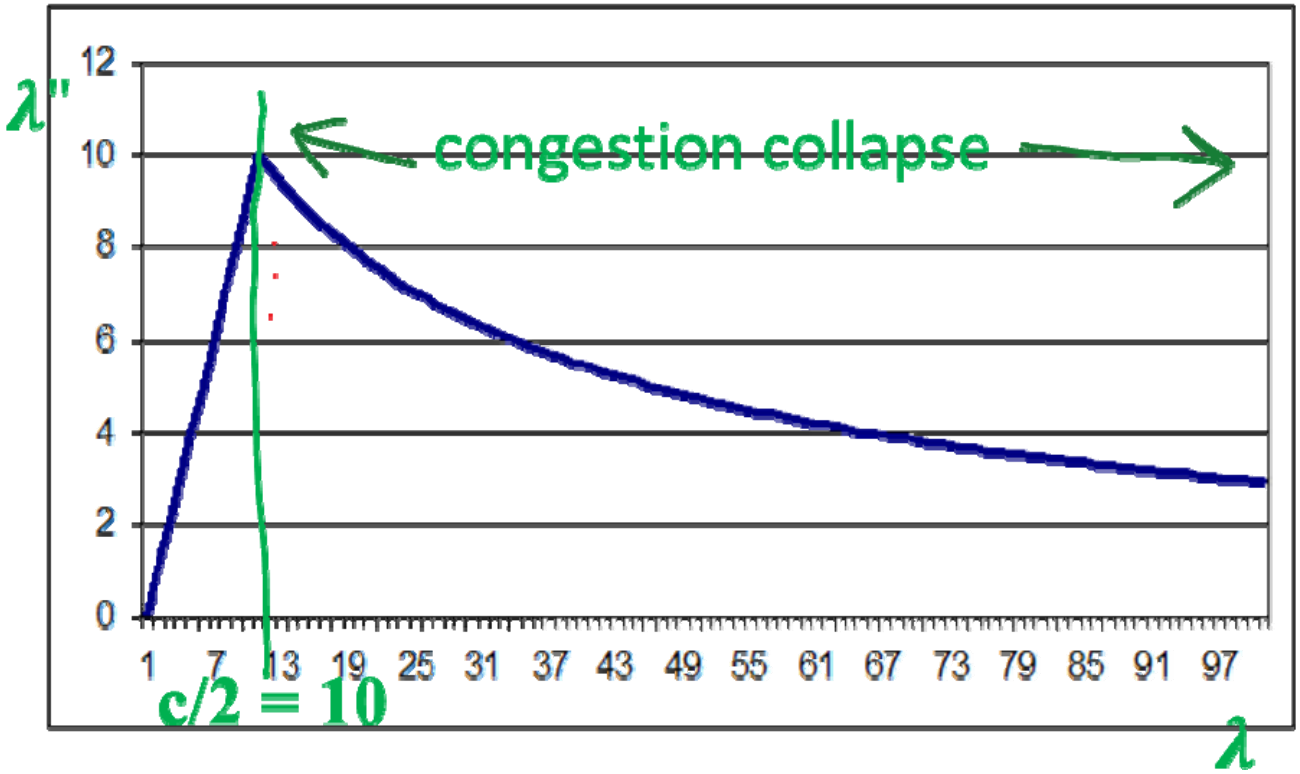


Another pattern…

One UDP source at every node, 2-hop flow, circular symmetry

For large offered load λ, what happens ?


source i
λ_i
node i
link i
node i+1
link (i-1)
λ_i'
link (i+1)
λ_i''
λ_i''


λ''
12
10
8
6
4
2
0
congestion collapse
1 7 13 19 25 31 37 43 49 55 61 67 73 79 85 91 97
c/2 = 10
λ

Congestion Collapse

Definition: Offered load increases, work done decreases

Frequent in complex systems

May be due to

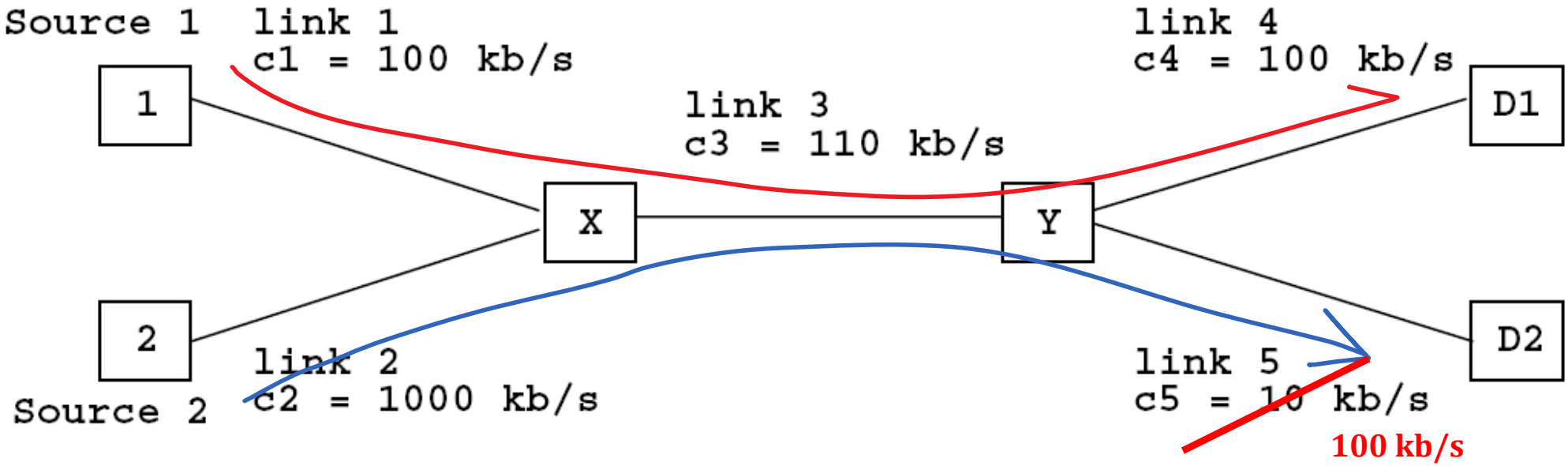
- cost per job increases with load
- Impatience (jobs leave before completion)
- Rejection of jobs before completion

Designer must do something to avoid congestion collapse

- Eg. Admission control in web servers
- Eg. TCP congestion control

Analyst must look for congestion collapse

Sources use TCP (= fair scheduling). Increase capacity of link 5 to 100 kb/s; what happens to source 1 ?



A. Its rate increases
B. Its rate decreases
C. Nothing happens
D. I don't know

Competition Side Effect

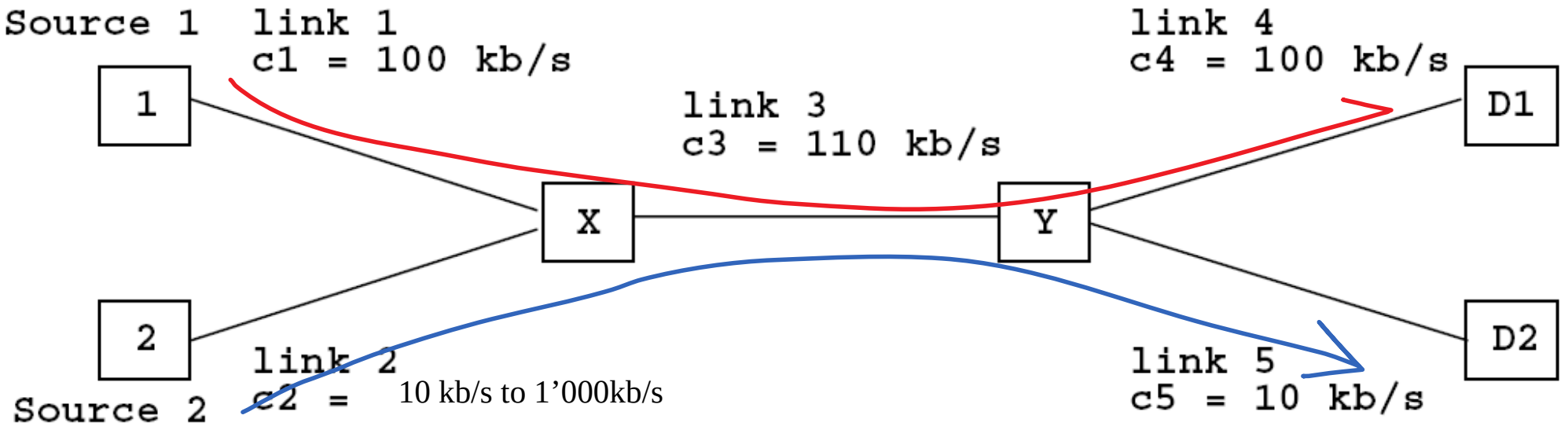
System balances resources according to some scheduling

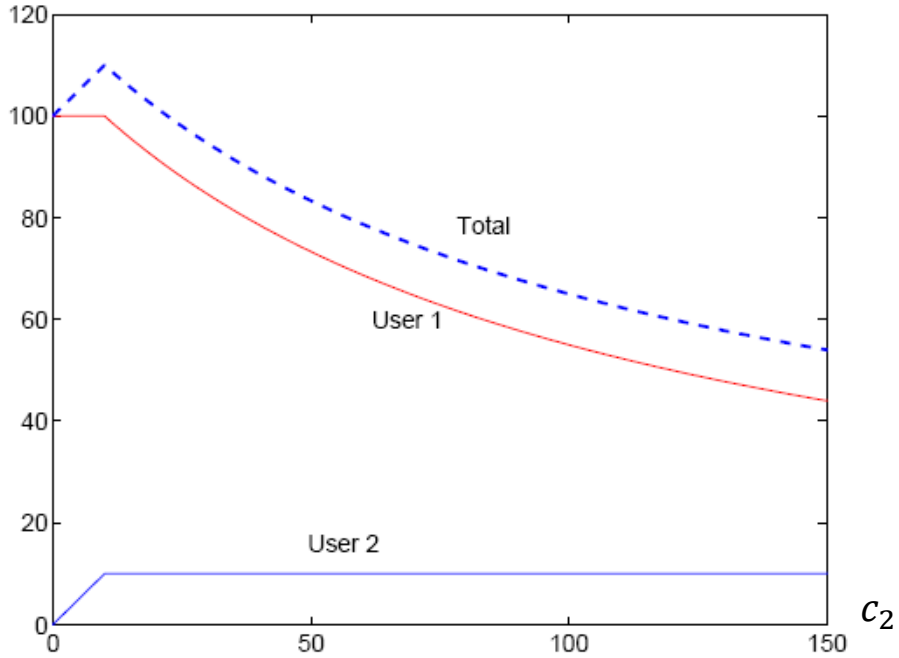
Putting more resources changes the outcome of the scheduling

Apparent paradox: put more resources, some get less

No TCP, users send as much can

Increase capacity of link 2 from 10 to 1000 kb/s





Latent Congestion Collapse

System is susceptible to congestion collapse

Low speed access prevents congestion collapse

Adding resources reveals congestion collapse

Apparent paradox: put more resources, all get less

Take Home Message

Watch for patterns, they are very frequent

- Bottlenecks
- Congestion collapse
- Competition side effects
- Latent Congestion collapse

Now it's your turn…

PERFORMANCE EVALUATION

HOMEWORK 1

1 ASSIGNMENT

Customers in Joe's shop are not satisfied because the downloading time is very large. Joe has hired you as performance analyst to understand the problem and propose some solutions.