On Some of My Simple Results

Leonard Kleinrock
Professor, UCLA Computer Science Dept.

Mobicom
September 8, 2014

©Leonard Kleinrock 2014

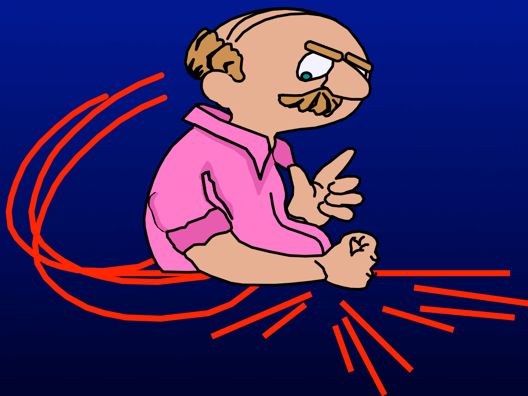
1. Data Traffic is Bursty and Asynchronous

1960's

©Leonard Kleinrock 2014

The Problem with Bursty Asynchronous Demands

- You cannot predict exactly **when** they will demand access
- You cannot predict **how much** they will demand
- Most of the time they **do not need** access
- When they ask for it, they want **immediate** access!!



©Leonard Kleinrock 2014

Conflict Resolution of Simultaneous Demands

- **Queueing:**
  - One gets served
  - All others wait
- **Splitting:**
  - Each gets a piece of the resource
- **Blocking:**
  - One gets served
  - All others are refused
- **Smashing:**
  - Nobody gets served !

A queueing system is a perfect resource sharing mechanism

It serves whatever work has arrived

©Leonard Kleinrock 2014

How Fast Can You Serve?

- Most queueing systems consider that the "server" can only work at the rate of **1 sec/sec**

©Leonard Kleinrock 2014

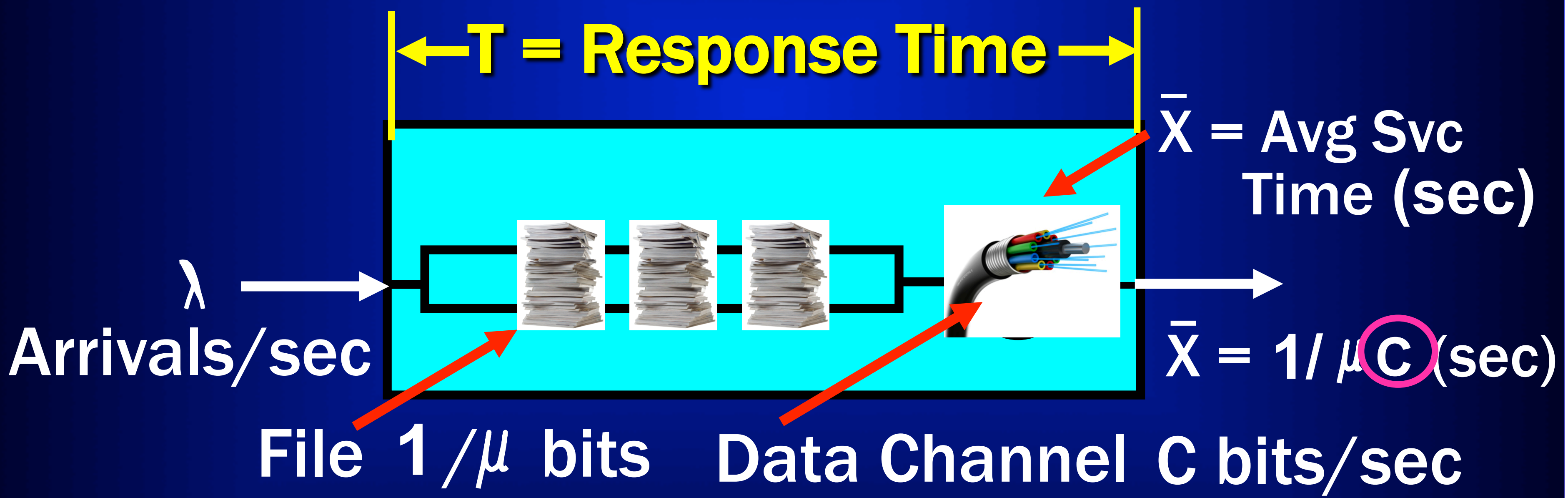
How Fast Can You Serve?

- Most queueing systems consider that the "server" can only work at the rate of **1 sec/sec**

©Leonard Kleinrock 2014

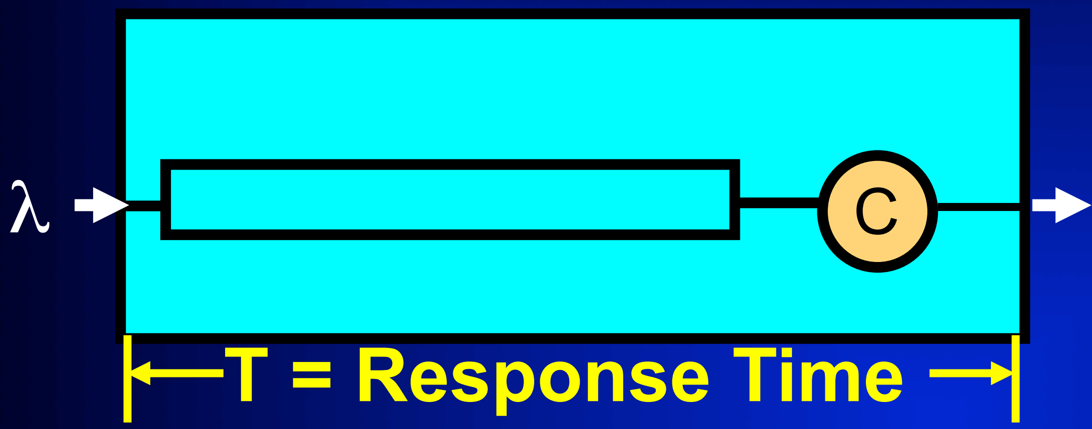
How Fast Can You Serve?

- Most queueing systems consider that the "server" can only work at the rate of 1 sec/sec
- Now replace humans with data technology



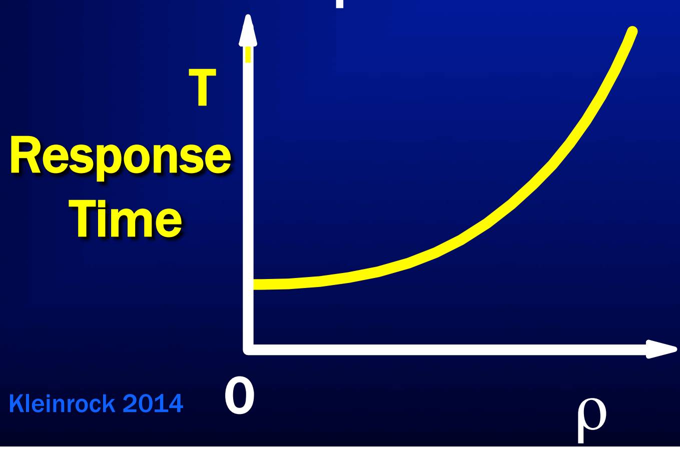
©Leonard Kleinrock 2014

The Basic M/M/1 Equation



- λ = Arrival rate (msg/sec)
- $1/\mu$ = Avg No. of bits/msg
- C = Capacity (bits/sec)
- $\bar{x} = 1/\mu C$ (sec)
- $\rho = \lambda \bar{x} = \lambda / \mu C$

$$T = \frac{\bar{x}}{1 - \rho} = \frac{\rho / \lambda}{1 - \rho}$$



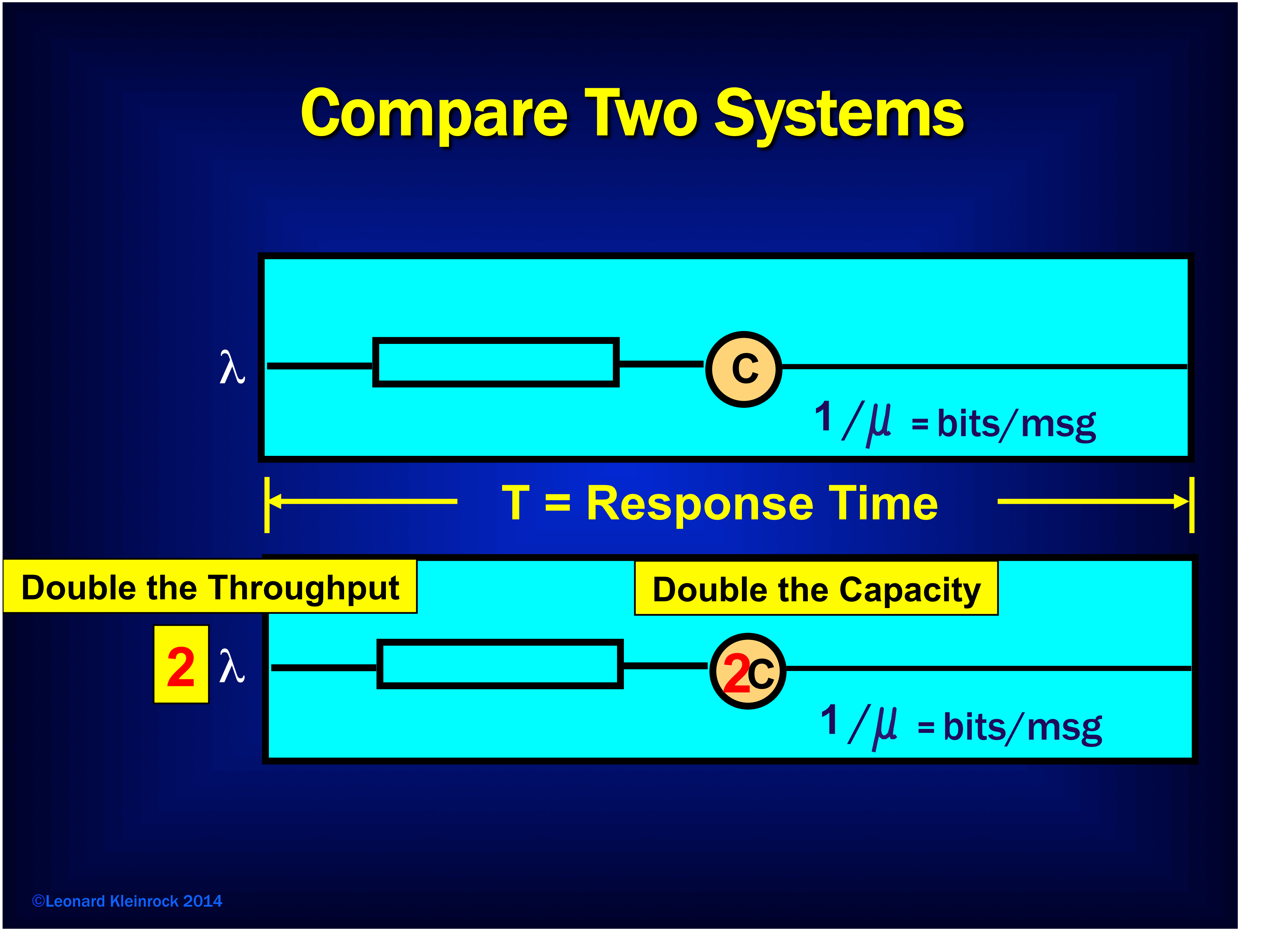
Now let's scale it up!

©Leonard Kleinrock 2014

2. Economy of Scale

1960's

©Leonard Kleinrock 2014



Compare Two Systems
λ
C
1/μ = bits/msg
T = Response Time
Double the Throughput
Double the Capacity
2 λ
2C
1/μ = bits/msg
©Leonard Kleinrock 2014

The Economy of Scale

- If you scale up throughput and capacity by some factor,

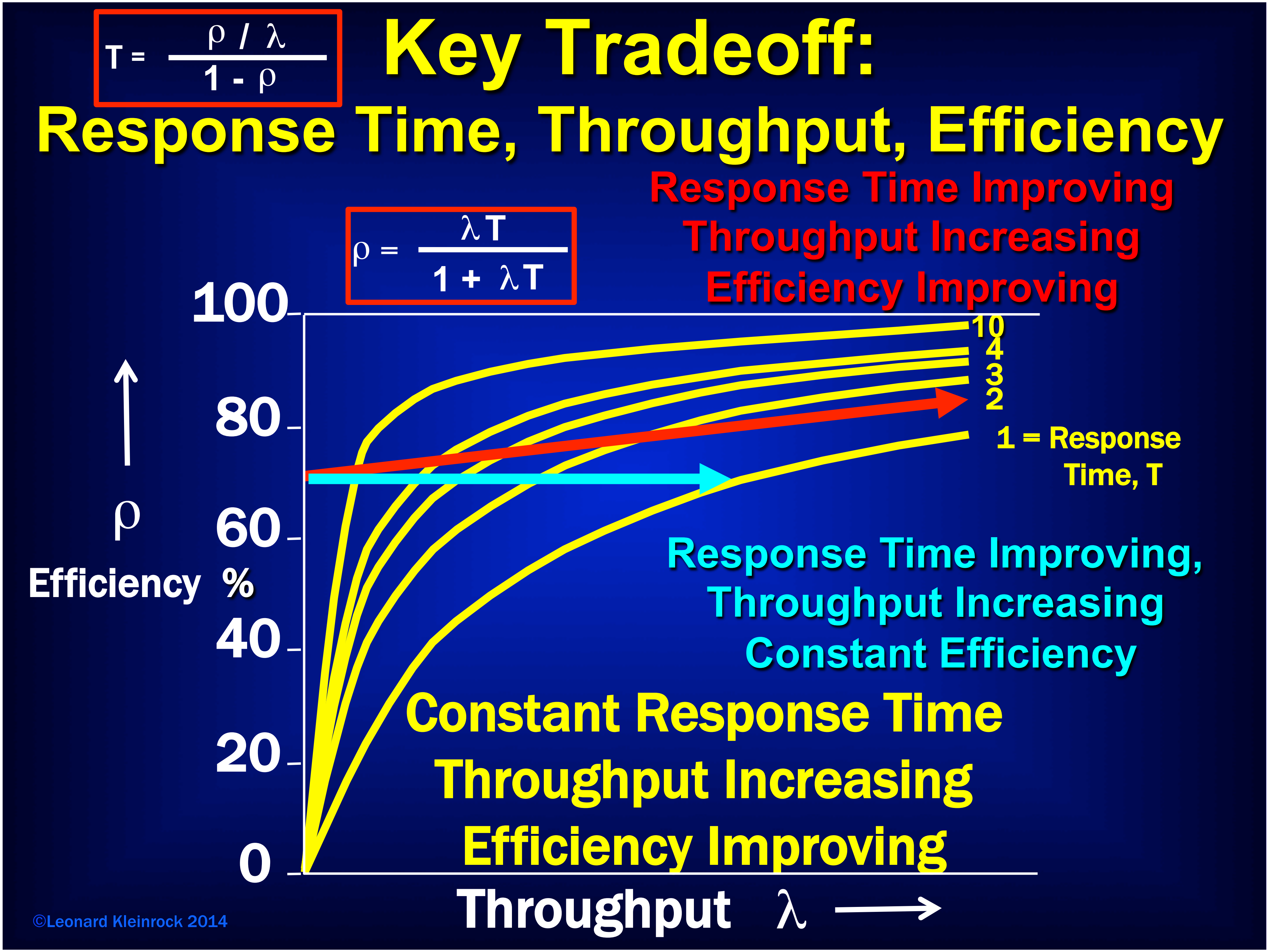
  *then you **reduce** response time by that same factor.*

- If you scale capacity more slowly than throughput while holding response time constant,

  then efficiency will increase (and can approach 100%).

- **If fact, you can improve all three!**

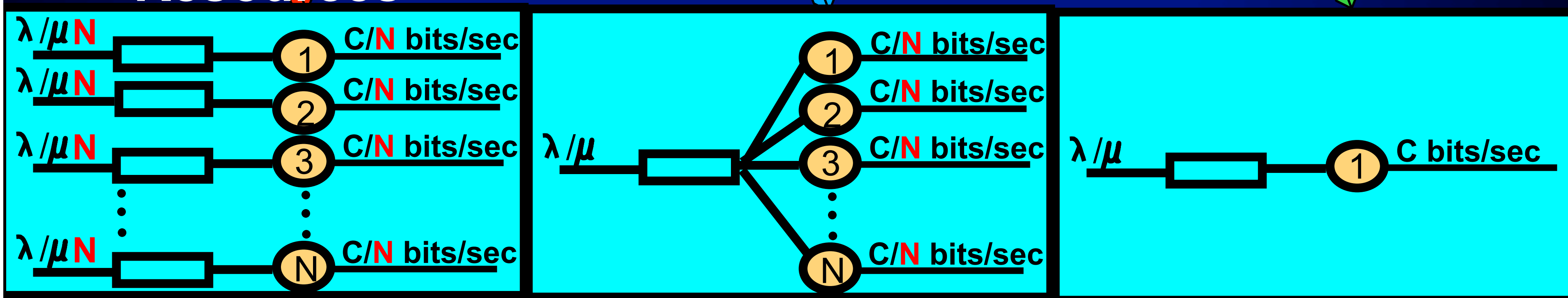
©Leonard Kleinrock 2014



T = ρ / λ over 1 - ρ
Key Tradeoff:
Response Time, Throughput, Efficiency
Response Time Improving
Throughput Increasing
Efficiency Improving
ρ = λT over 1 + λT
100
80
60
40
20
0
ρ
Efficiency %
10
4
3
2
1 = Response Time, T
Response Time Improving,
Throughput Increasing
Constant Efficiency
Constant Response Time
Throughput Increasing
Efficiency Improving
Throughput λ
©Leonard Kleinrock 2014

Comparing Architectures

Dedicated Resources | **Shared Resources** | **LARGE Shared Resources**

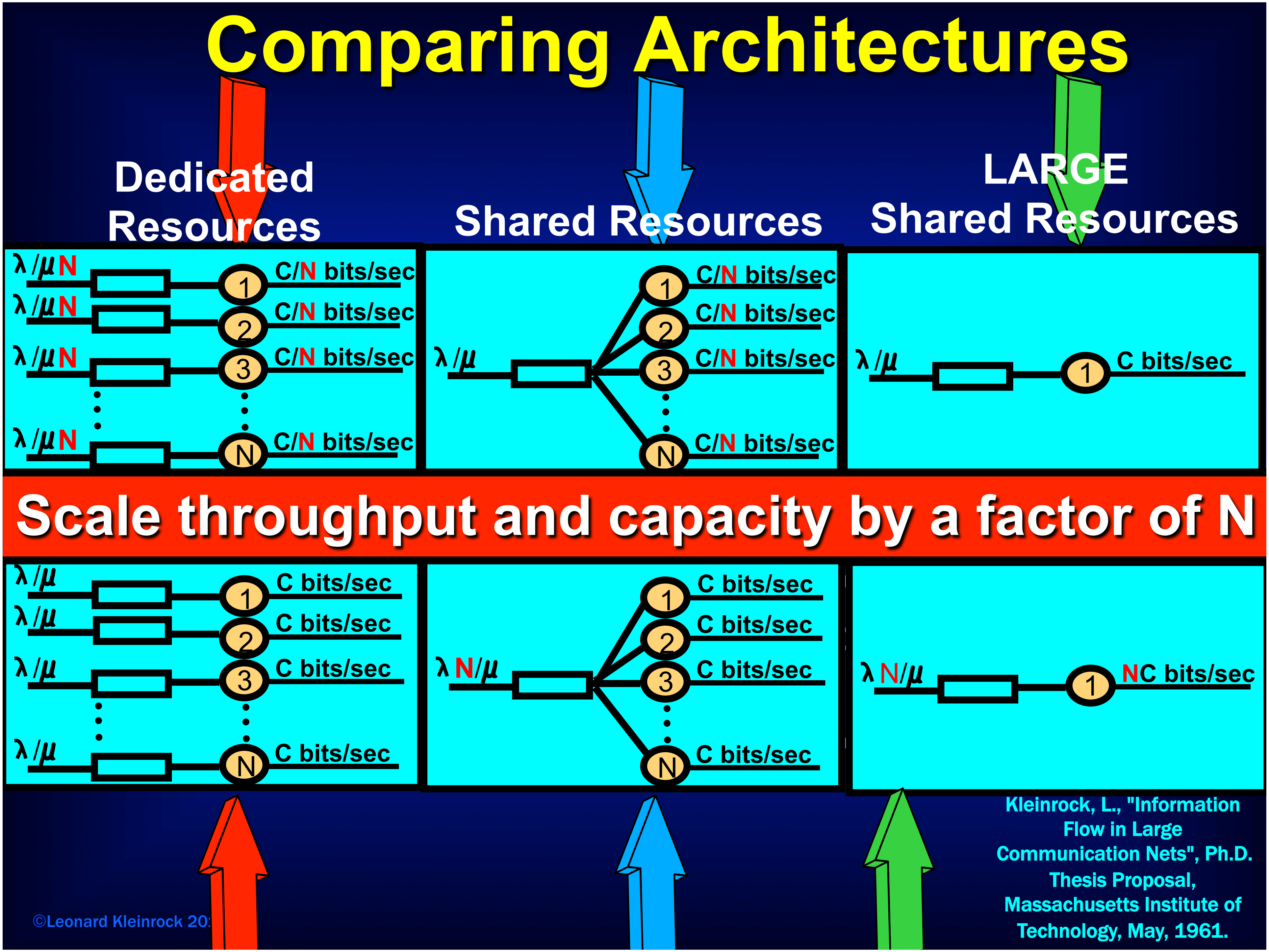


What is the optimum number of channels to minimize the mean response time?

Theorem: The optimum value of N which **minimizes** the mean response time through the switch is:

N=1

Kleinrock, L., "Information Flow in Large Communication Nets", Ph.D. Thesis Proposal, Massachusetts Institute of Technology, May, 1961.

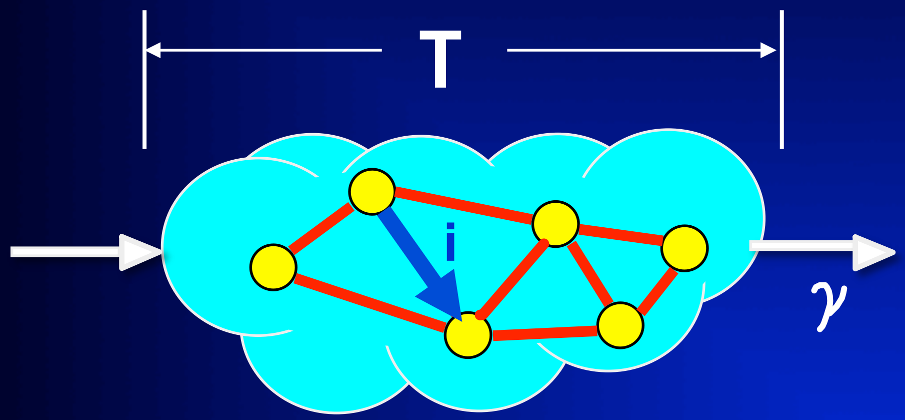

Comparing Architectures
Dedicated Resources
Shared Resources
LARGE Shared Resources
λ/μN
C/N bits/sec
1
2
3
N
λ/μ
C bits/sec
Scale throughput and capacity by a factor of N
λ/μ
C bits/sec
λN/μ
NC bits/sec
Kleinrock, L., "Information Flow in Large Communication Nets", Ph.D. Thesis Proposal, Massachusetts Institute of Technology, May, 1961.
©Leonard Kleinrock 20:

3. Data Networks

1960's

©Leonard Kleinrock 2014

Networks of Arbitrary Topology



$$T = \sum_i \frac{\lambda_i}{\gamma} T_i$$

T = Average network delay

λ_i = Traffic on channel i (Msg/sec)

γ = Network throughput (Msg/sec)

T_i = Average delay for channel i

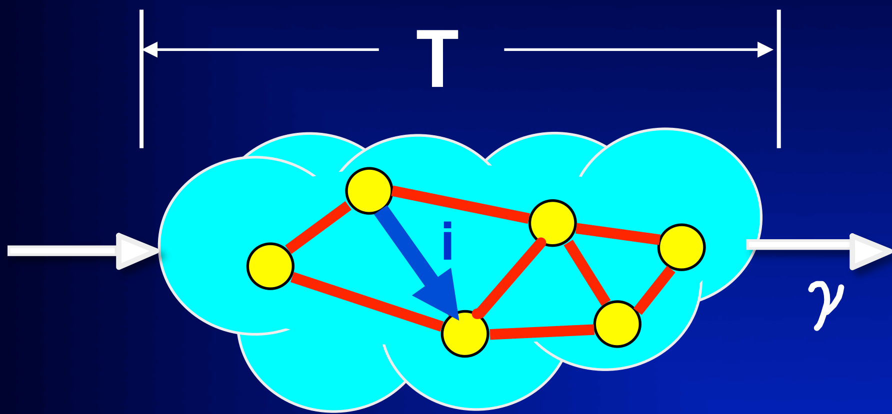
Key equation for network delay.

And it is EXACT!!

Kleinrock, L., "*Message Delay in Communication Nets with Storage*" (Ph.D. thesis, MIT, December 1962)

©Leonard Kleinrock 2014

Proof



$\gamma T = \bar{N}$ Little's Result for the full network

$\bar{N} = \sum \bar{N}_i$

$\lambda_i T_i = \bar{N}_i$ Little's Result for each channel

$\gamma T = \sum \lambda_i T_i$

$$T = \sum_i \frac{\lambda_i}{\gamma} T_i$$

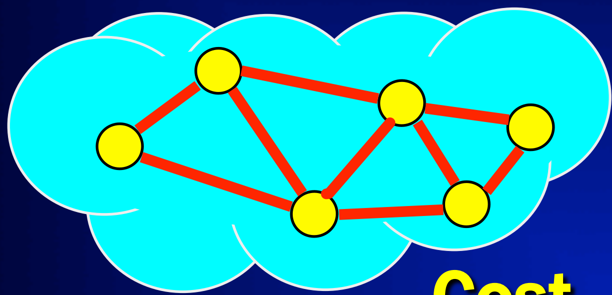
©Leonard Kleinrock 2014

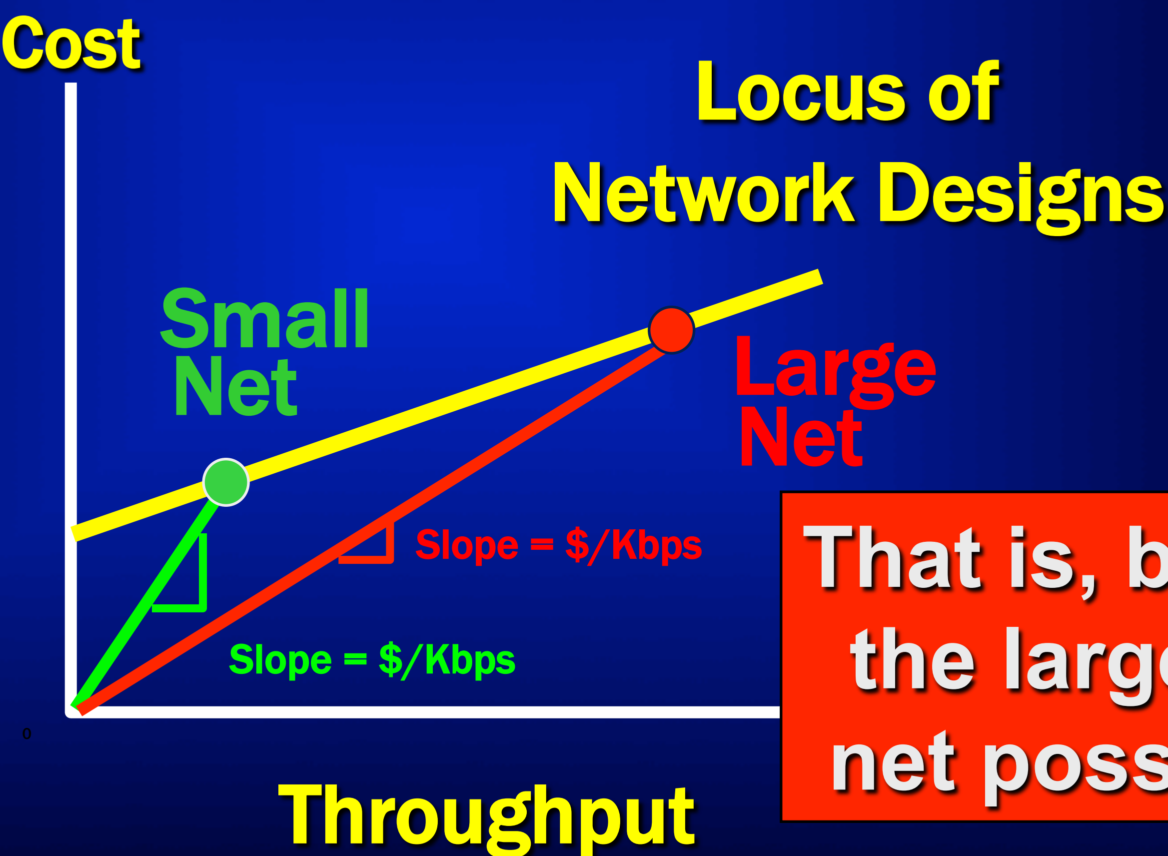
The Underlying Principles

- Resource Sharing (demand access)
  - Only assign a resource to data that is present
  - Examples are:
    - Message switching
    - Packet switching
    - Polling
    - ATDM
- Economy of Scale in Networks
  - Bigger is better
- Distributed control
  - It is efficient, stable, robust, fault-tolerant and WORKS!

©Leonard Kleinrock 2014

Economy of Scale in Networks: Cost





That is, build the largest net possible

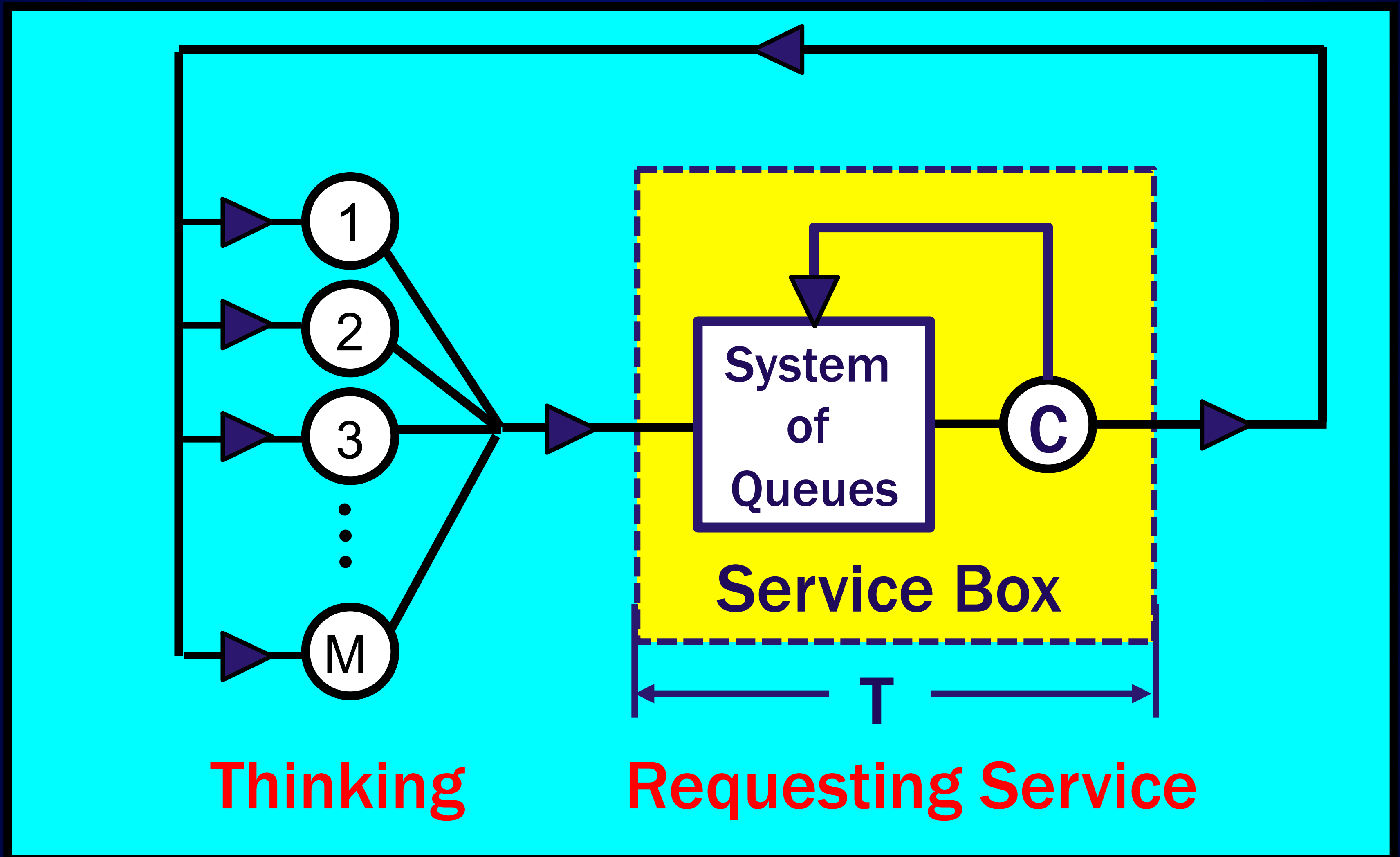
©Leonard Kleinrock 2014

4. Finite Population Models

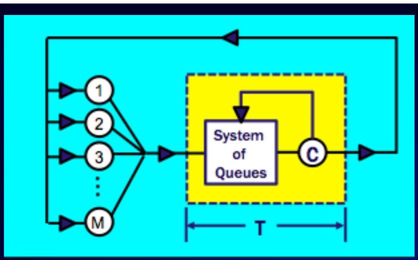
Late 1960's

©Leonard Kleinrock 2014

Finite Population Models



©Leonard Kleinrock 2014



Finite Population Models

M = Number of jobs (population size)

λ = Rate of job requests/thinking job

$1/\lambda$ = Average think time per thinking job

T = Average Response time in "Service Box"

τ = Cycle Time = $1/\lambda$ + T

Input rate of jobs to system = $M\lambda \dfrac{1/\lambda}{1/\lambda + T}$

$1/\mu$ = Avg No. of opns/job ($1/\mu C$ sec)

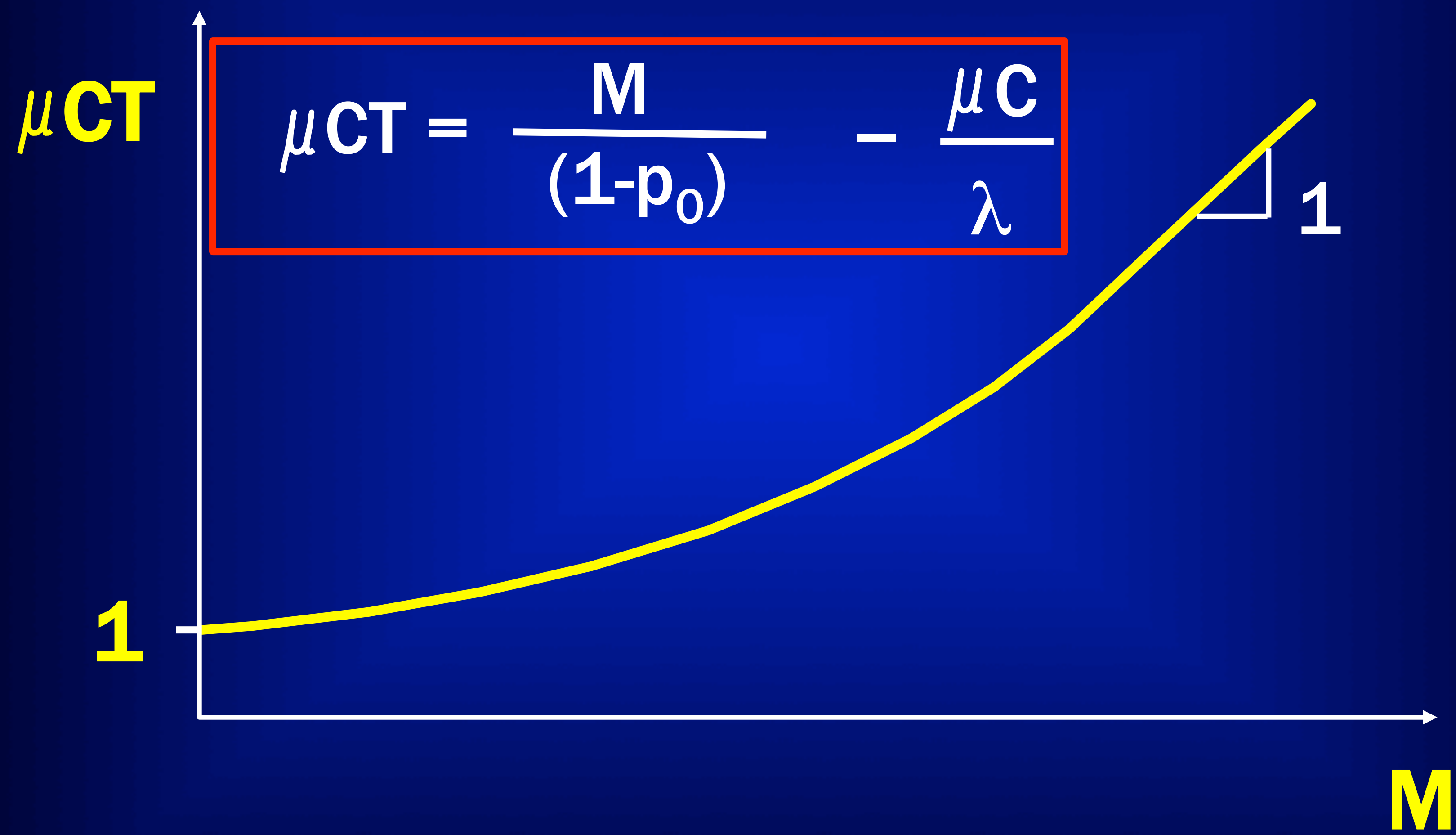
Output rate of jobs from system = $\mu C\,(1-p_0)$

$$\mu CT = \frac{M}{(1-p_0)} - \frac{\mu C}{\lambda}$$

©Leonard Kleinrock 2014

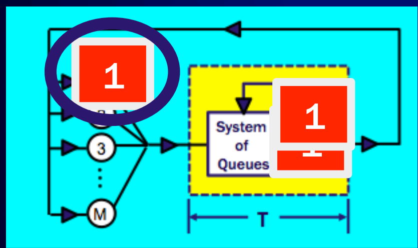
Finite Population Models

$$\mu CT = \frac{M}{(1-p_0)} - \frac{\mu C}{\lambda}$$



©Leonard Kleinrock 2014

Deterministic Model

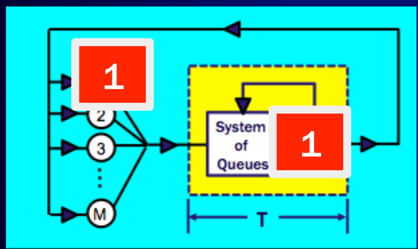


Suppose each job takes exactly $1/\lambda$ sec thinking

Suppose each job needs exactly $1/\mu C$ sec of service

©Leonard Kleinrock 2014

Deterministic Model



Suppose each job takes **exactly** $1/\lambda$ sec thinking

Suppose each job needs **exactly** $1/\mu C$ sec of service

Now add one more job! And another job!

©Leonard Kleinrock 2014

The "Saturation" Point

- Looks like we just filled the system with 6 carefully placed deterministic jobs.
- In general, without interference of jobs, for this deterministic system, we could fit exactly

$$M^* = \frac{1/\lambda + 1/\mu C}{1/\mu C} = \frac{\lambda + \mu C}{\lambda}$$

- Let's define this number as the saturation number, M*

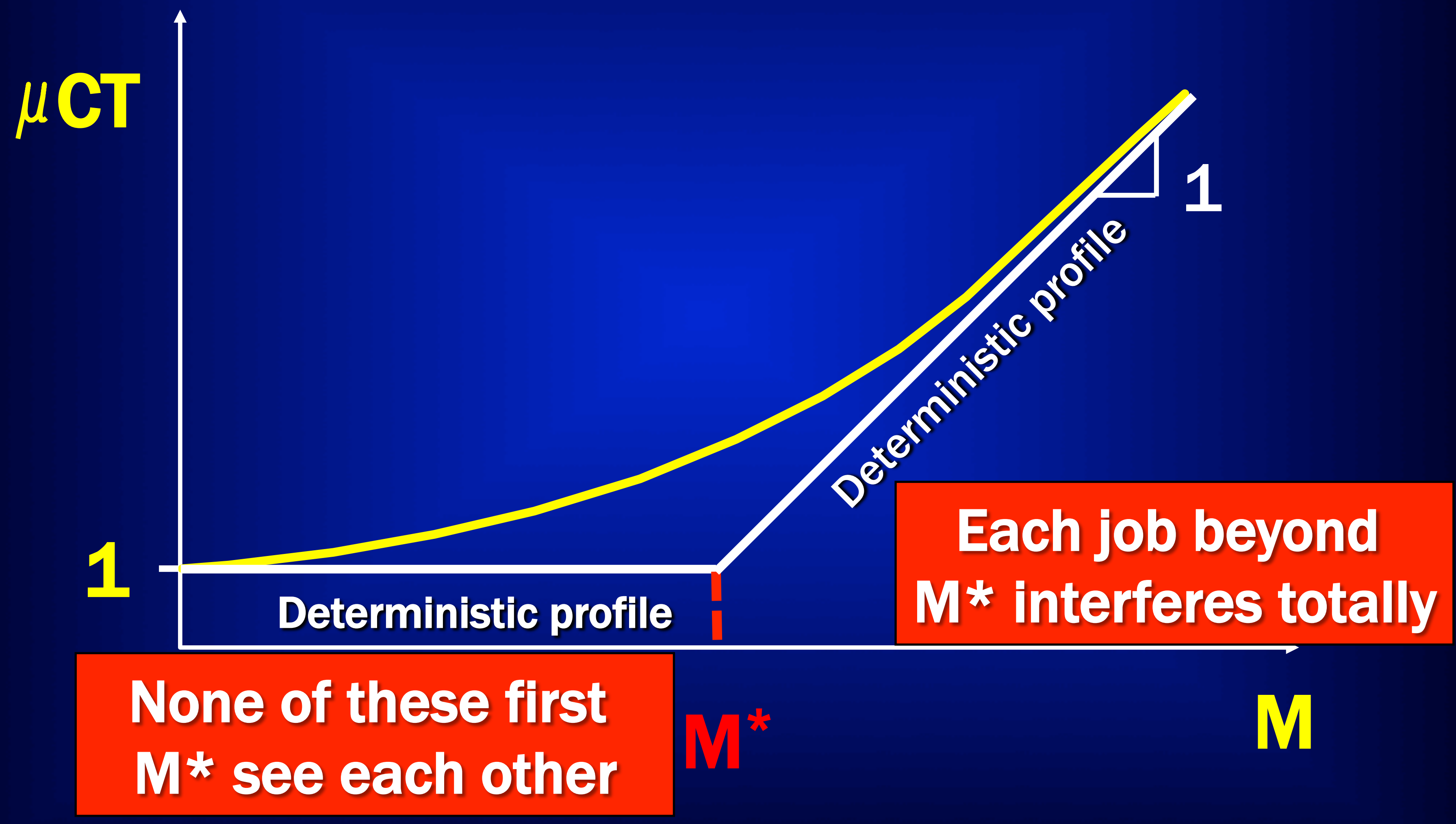
Thus we can fit M* jobs in and they don't see each other

The first M* jobs look just like 1 job

L. Kleinrock, "Certain Analytic Results for Time-Shared Processors," in Proceedings of the International Federation for Information Processing Congress, Edinburg, Scotland, August 1968, p. d119–d125.

©Leonard Kleinrock 2014

The Deterministic Profile



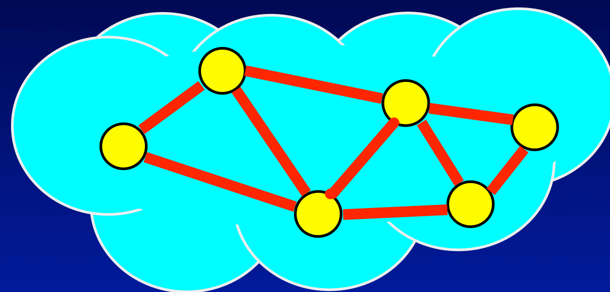
©Leonard Kleinrock 2014

5. Flow Control and “Power”

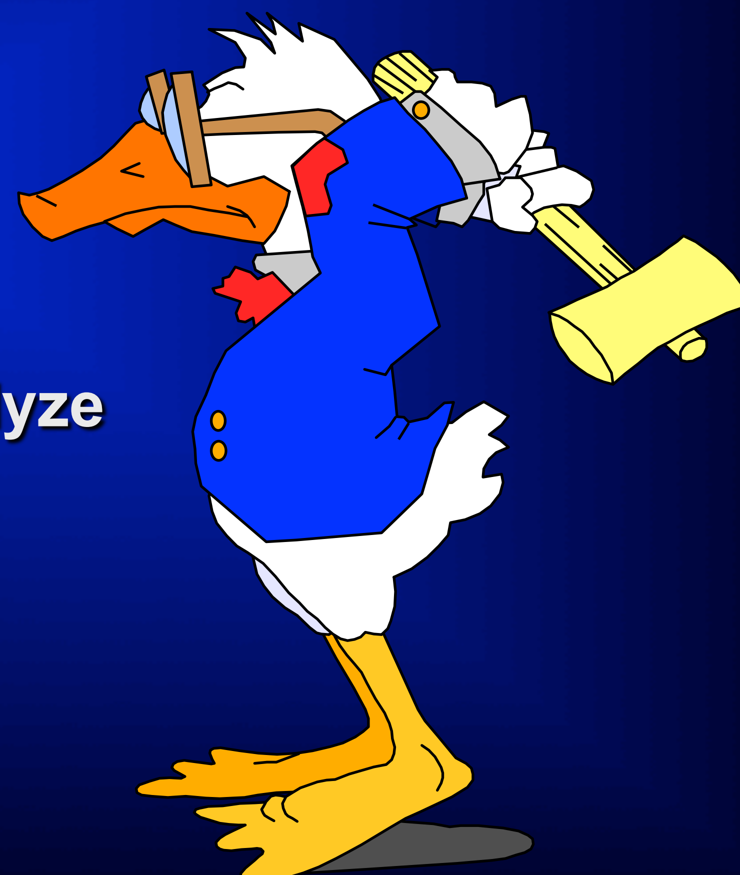
1970’s

©Leonard Kleinrock 2014

Flow Control Issues



- Routing Procedures:
  - Easy to design
  - Hard to analyze (dynamic)
- Flow Control:
  - Hard to design
  - Outrageously difficult to analyze
  - Absolutely essential
  - **Guaranteed to GET you!**

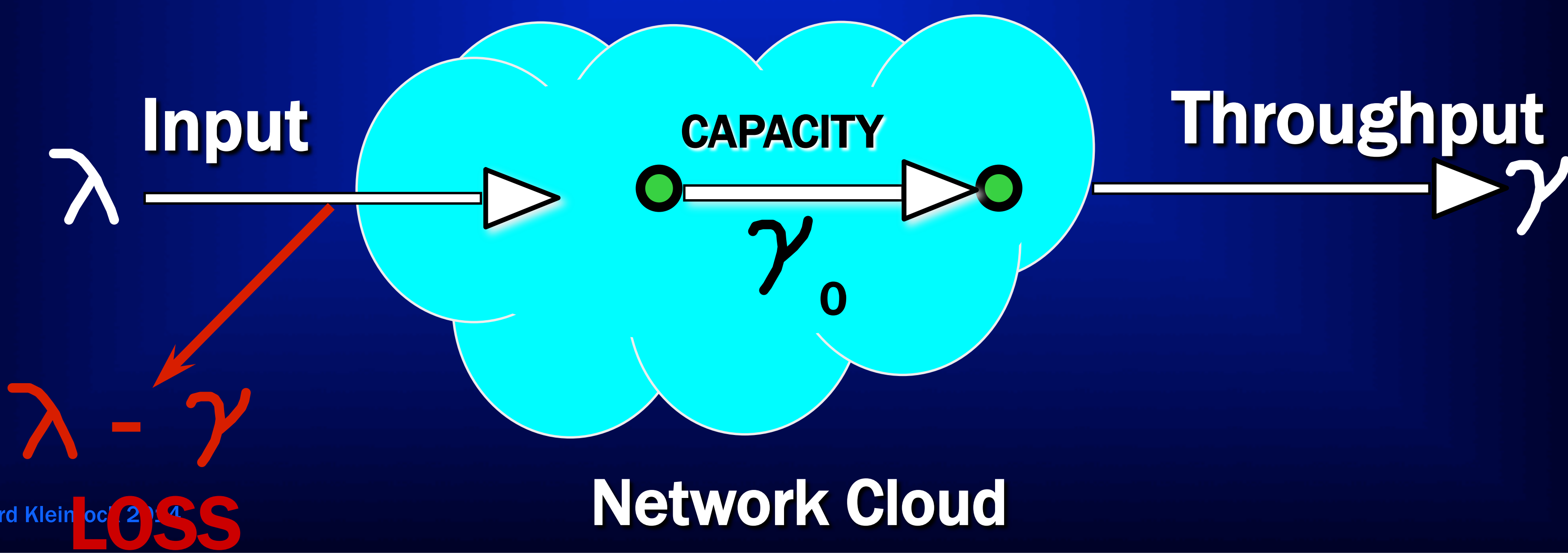


©Leonard Kleinrock 2014

Flow Control in Networks

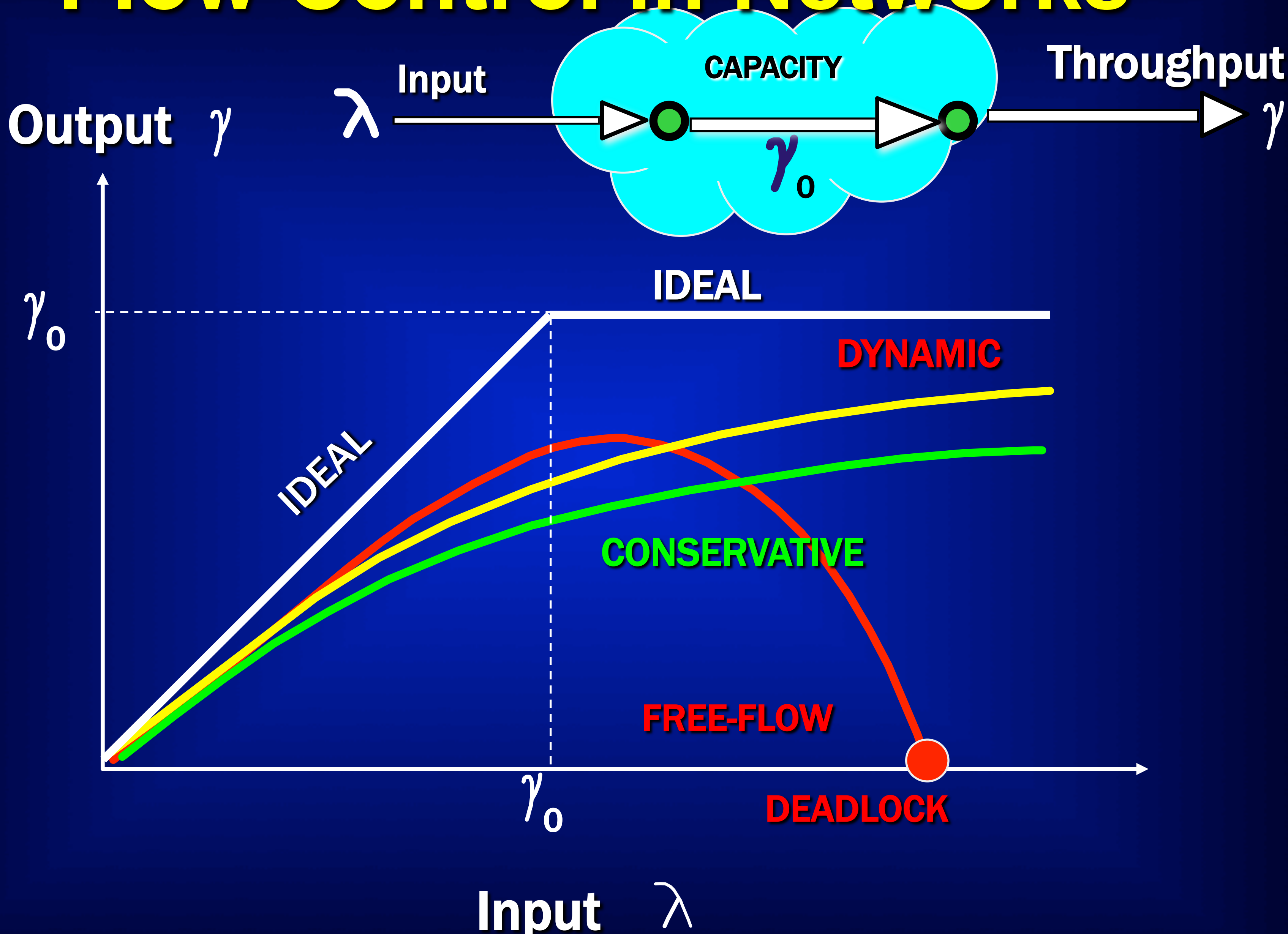
Throughput

Loss



©Leonard Kleinrock 2014

Flow Control in Networks

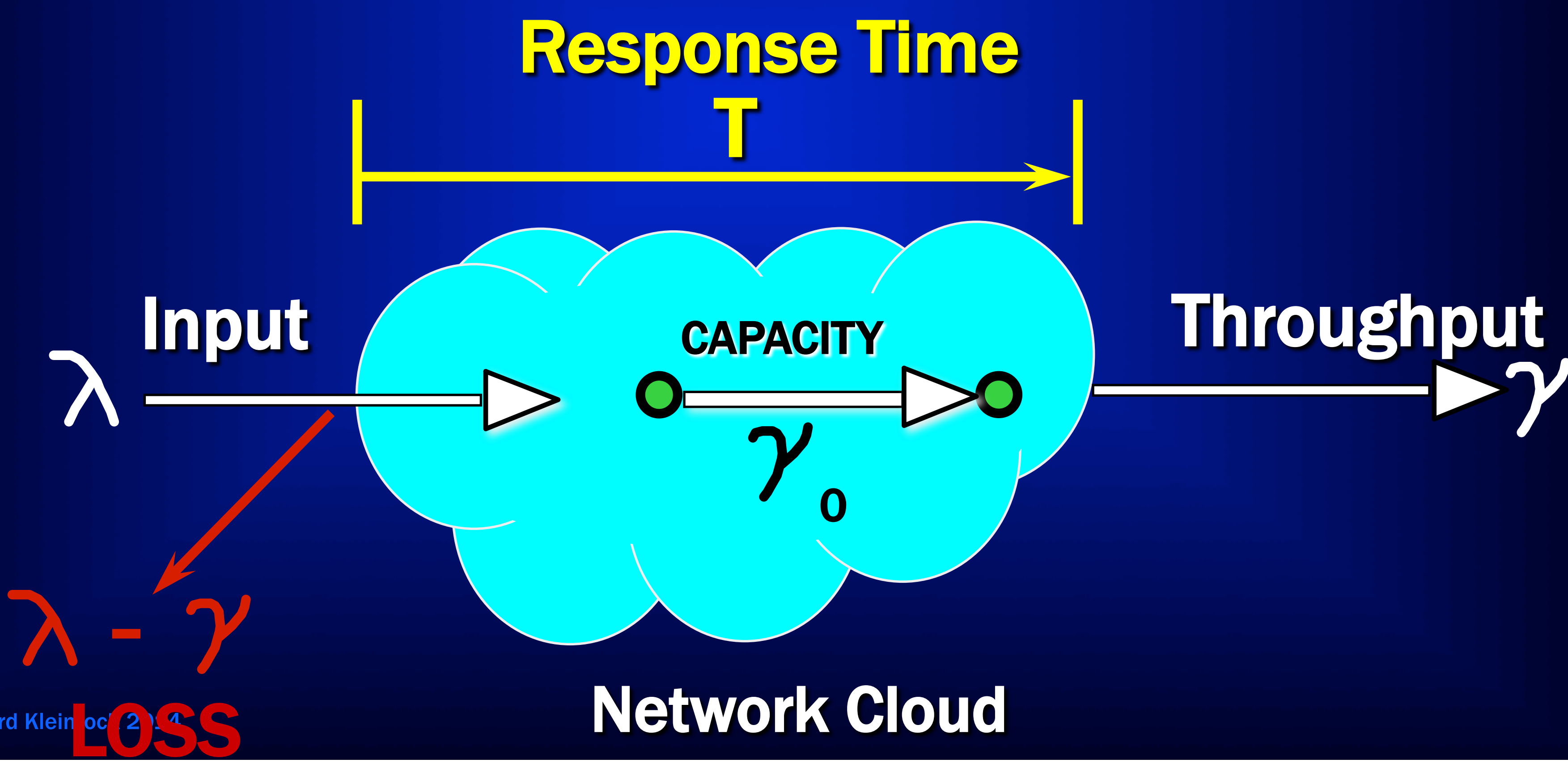


©Leonard Kleinrock 2014

Response Time

Throughput

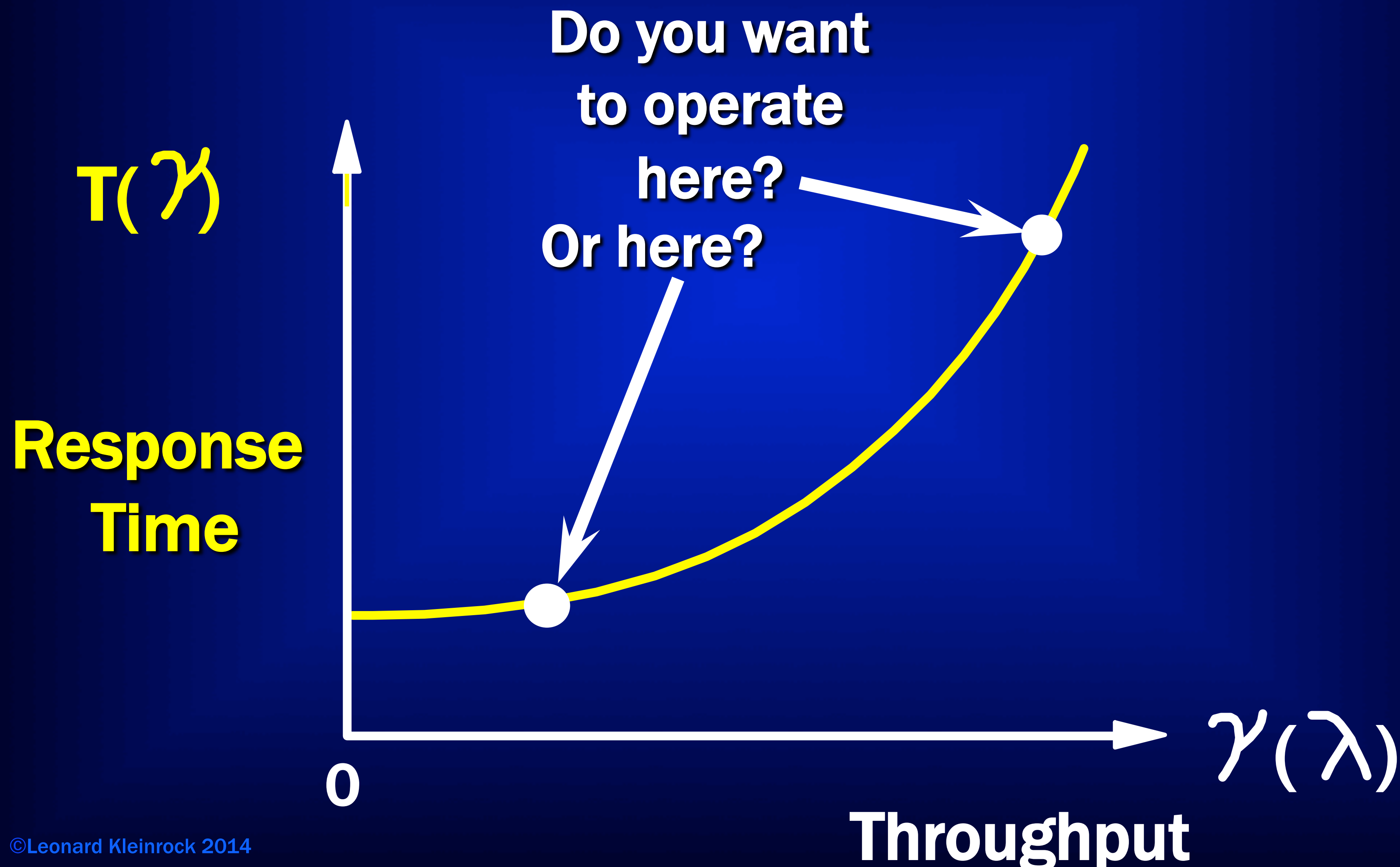
Loss



©Leonard Kleinrock 2014

Response Time vs Throughput

Now let's ask a good question:



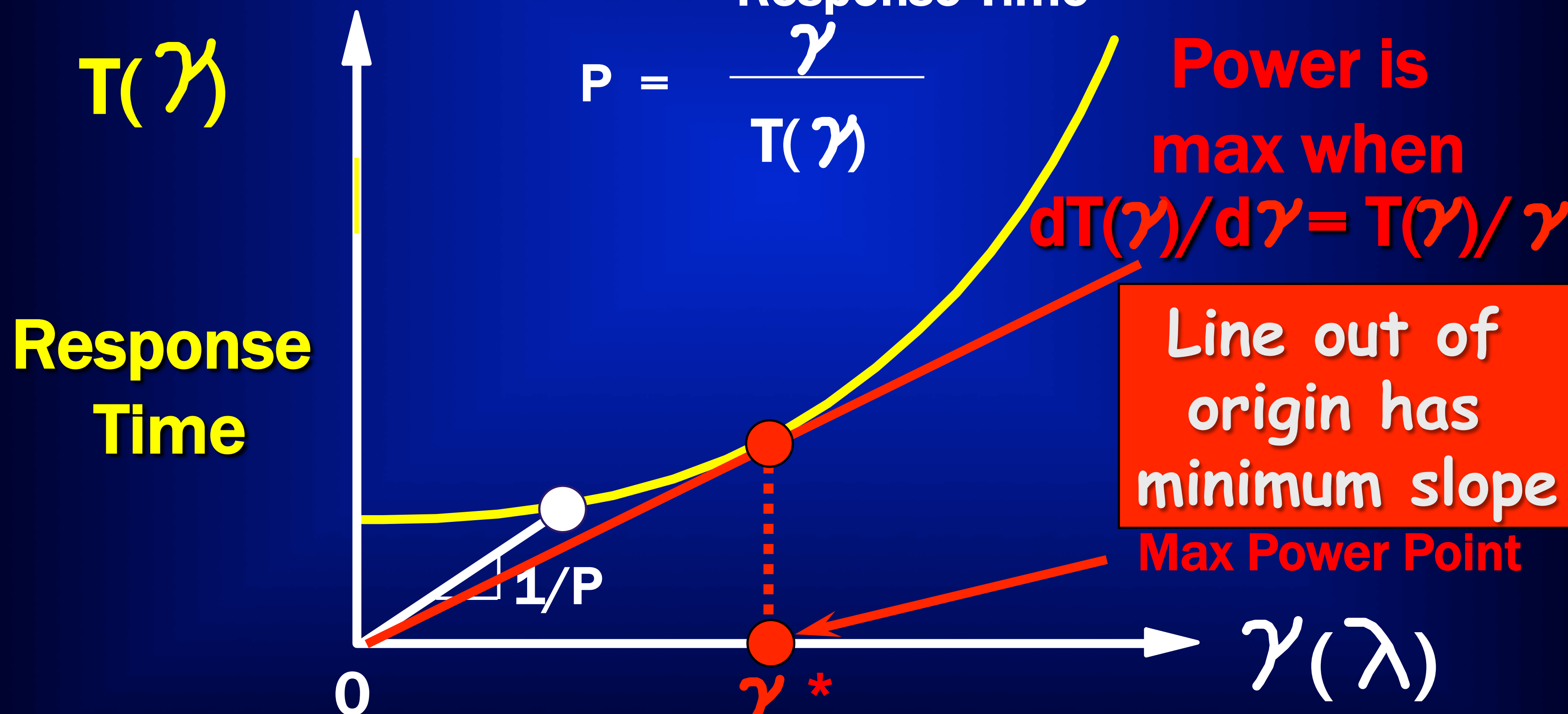
©Leonard Kleinrock 2014

Response Time vs Throughput

Let's define a new metric of performance:

$$\text{POWER} \triangleq \frac{\text{Throughput}}{\text{Response Time}}$$

$$P = \frac{\gamma}{T(\gamma)}$$



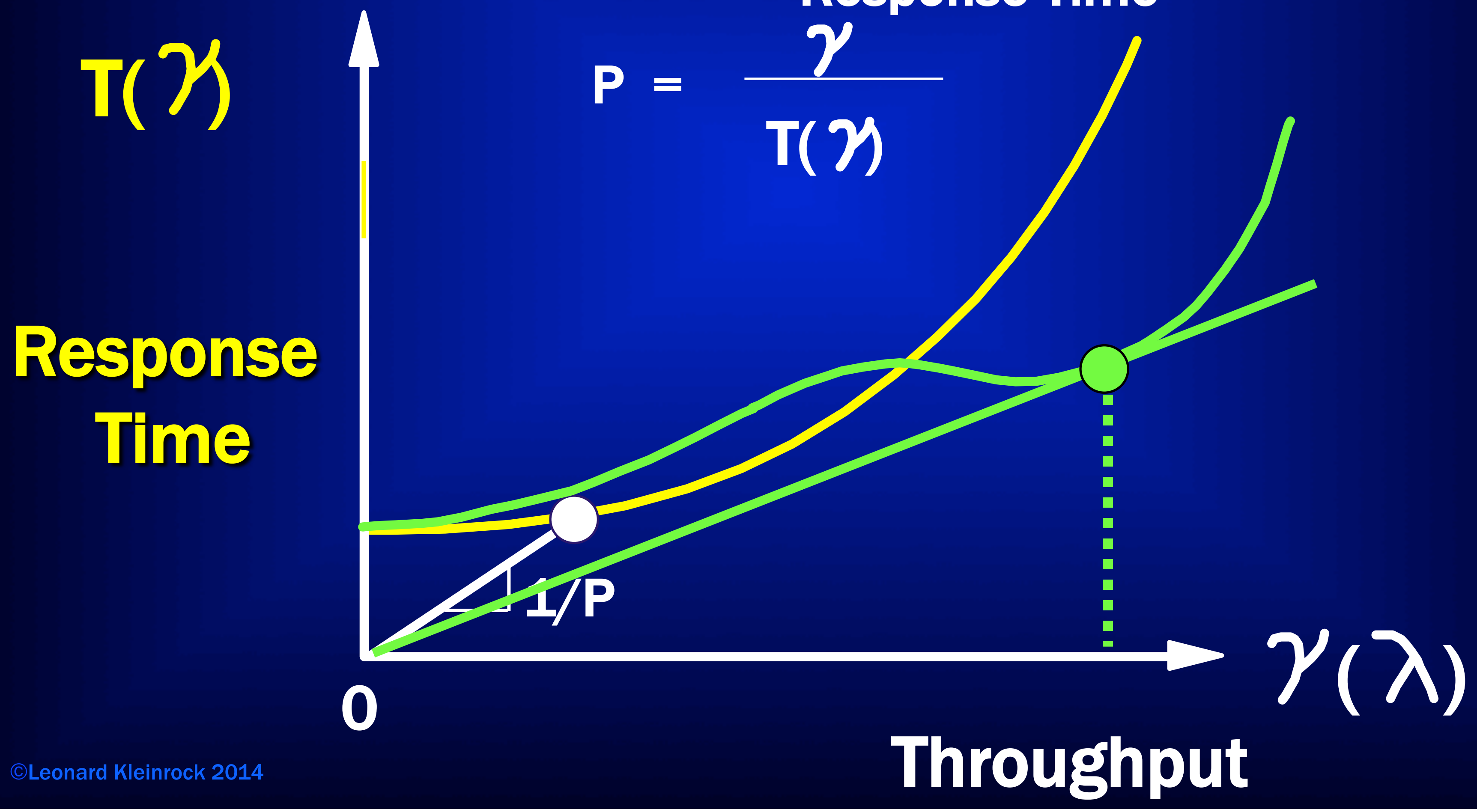
Kleinrock, L., "On Flow Control in Computer Networks", Conference Record, *Proceedings of the International Conference on Communications*, Vol. II, Toronto, Ontario, pp. 27.2.1 to 27.2.5, June 1978.

Response Time vs Throughput

We need a new metric of performance:

$$\text{POWER} \triangleq \frac{\text{Throughput}}{\text{Response Time}}$$

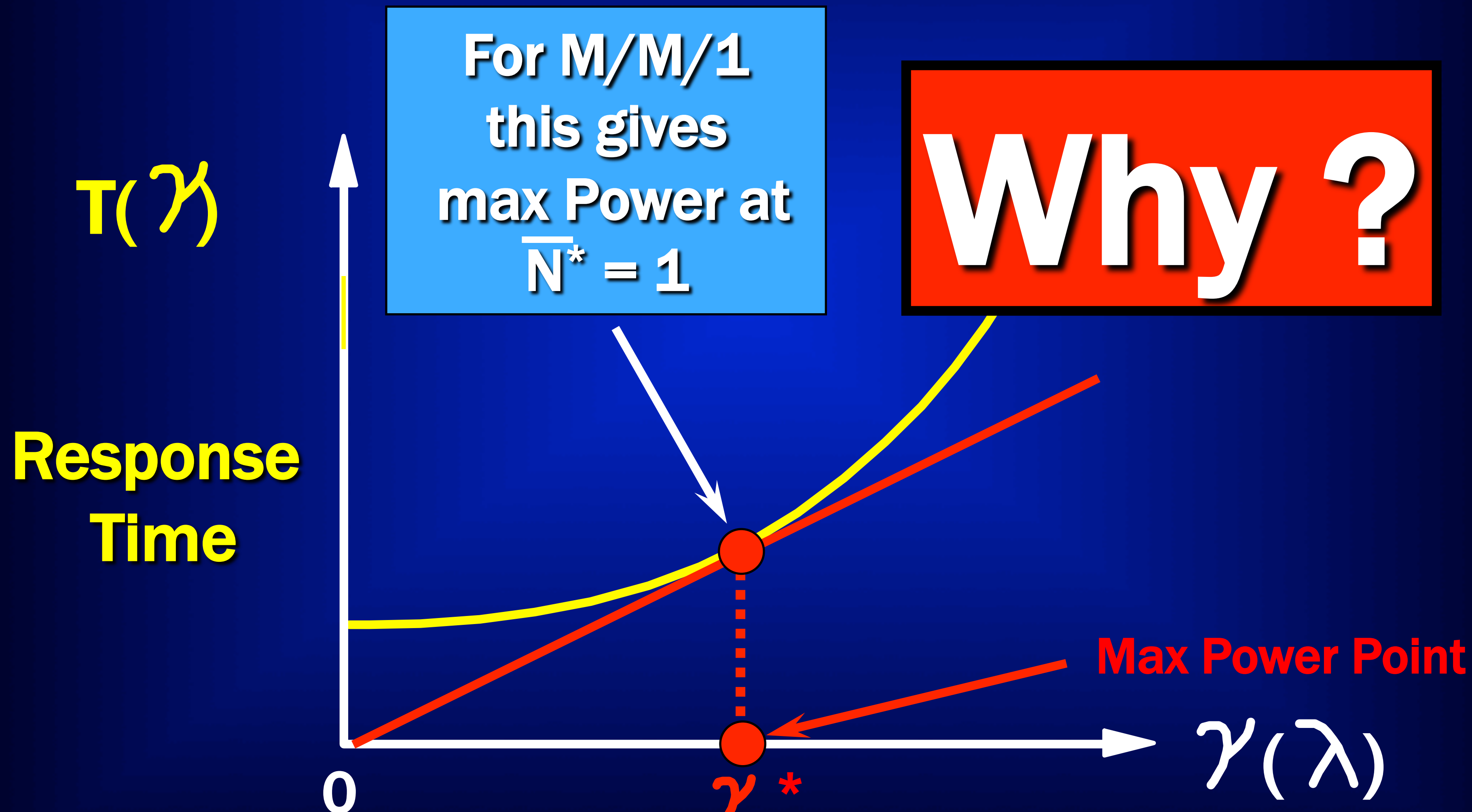
$$P = \frac{\gamma}{T(\gamma)}$$



©Leonard Kleinrock 2014

Response Time vs Throughput

Let's Dig Deeper on Understanding

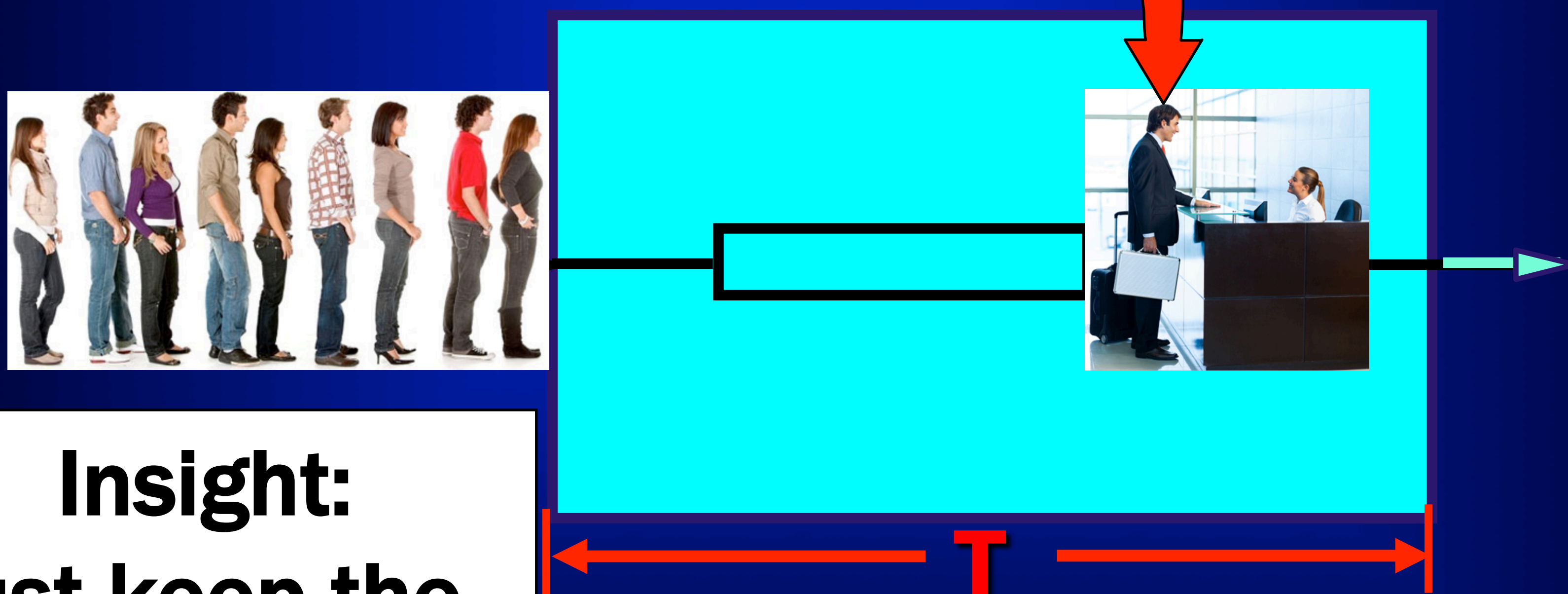


Kleinrock, L., "On Flow Control in Computer Networks", Conference Record, *Proceedings of the International Conference on Communications*, Vol. II, Toronto, Ontario, pp. 27.2.1 to 27.2.5, June 1978.

Understand Your Own Results

Use Your Intuition

Only 1 customer in the system



T = Min
Eff = Max

Insight:
Just keep the pipe full!

©Leonard Kleinrock 2014

Understand Your Own Results

- Our intuition says put exactly one person in the queueing system
  - This was from "deterministic" reasoning.
- We can't actually do that in general
- BUT our earlier result said that we should adjust the system to achieve an average of one person in the queueing system, i.e.,

At Max Power
$\overline{N}^* = 1$
for M/M/1

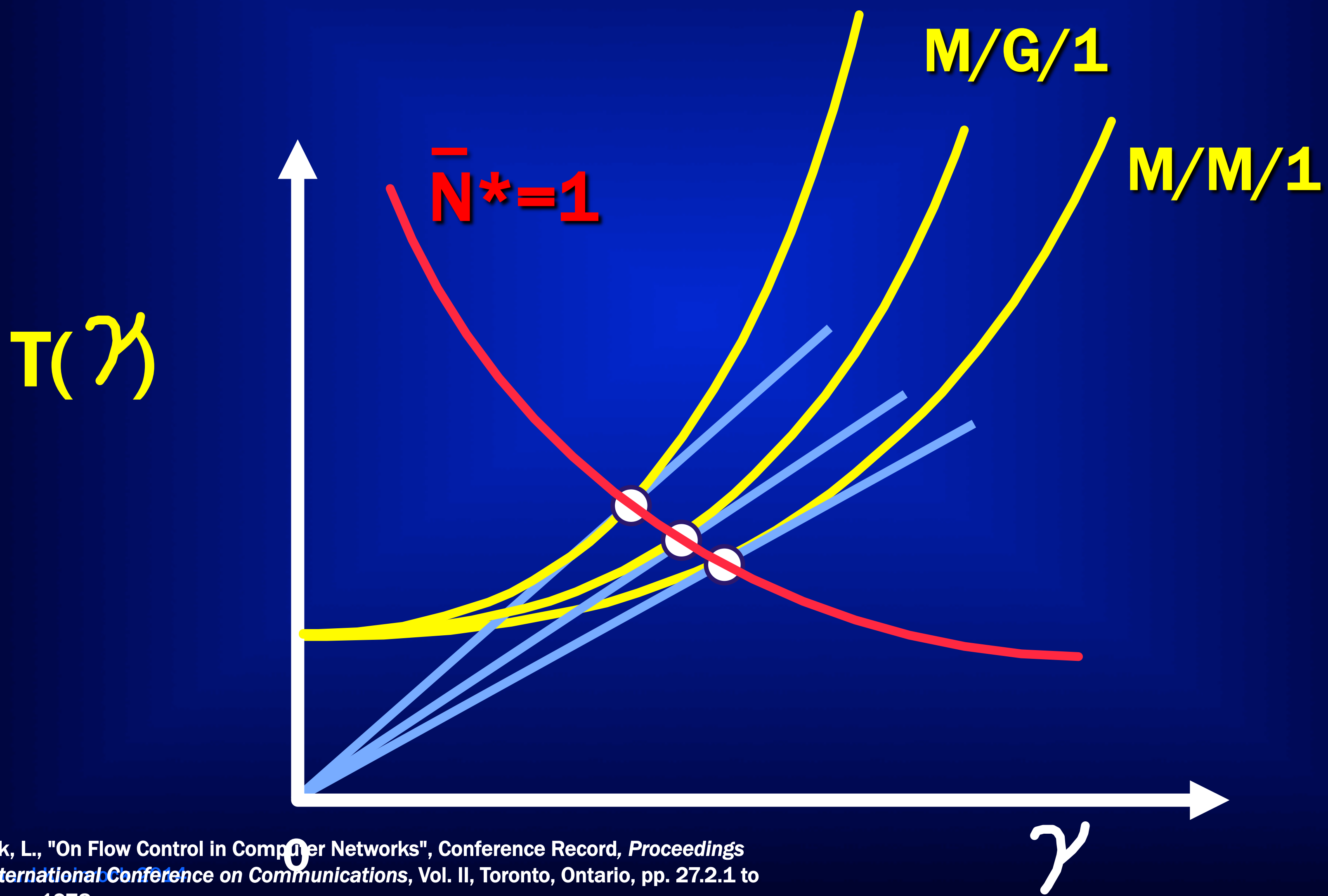
Further:
At Max Power we get
- **½ maximum thpt**
- **2x minimum delay**

for M/M/1

©Leonard Kleinrock 2014

Gee, that's funny!

What can we say for M/G/1 ?



Kleinrock, L., "On Flow Control in Computer Networks", Conference Record, *Proceedings of the International Conference on Communications*, Vol. II, Toronto, Ontario, pp. 27.2.1 to 27.2.5, June 1978.

A More General Power Definition

$$\text{POWER} \triangleq \frac{(\text{Throughput})^{r}}{\text{Response Time}}$$

$$P = \frac{\gamma^{r}}{T(\gamma)}$$

At Max Power

$$\overline{N}^{*} = r$$

for M/M/1

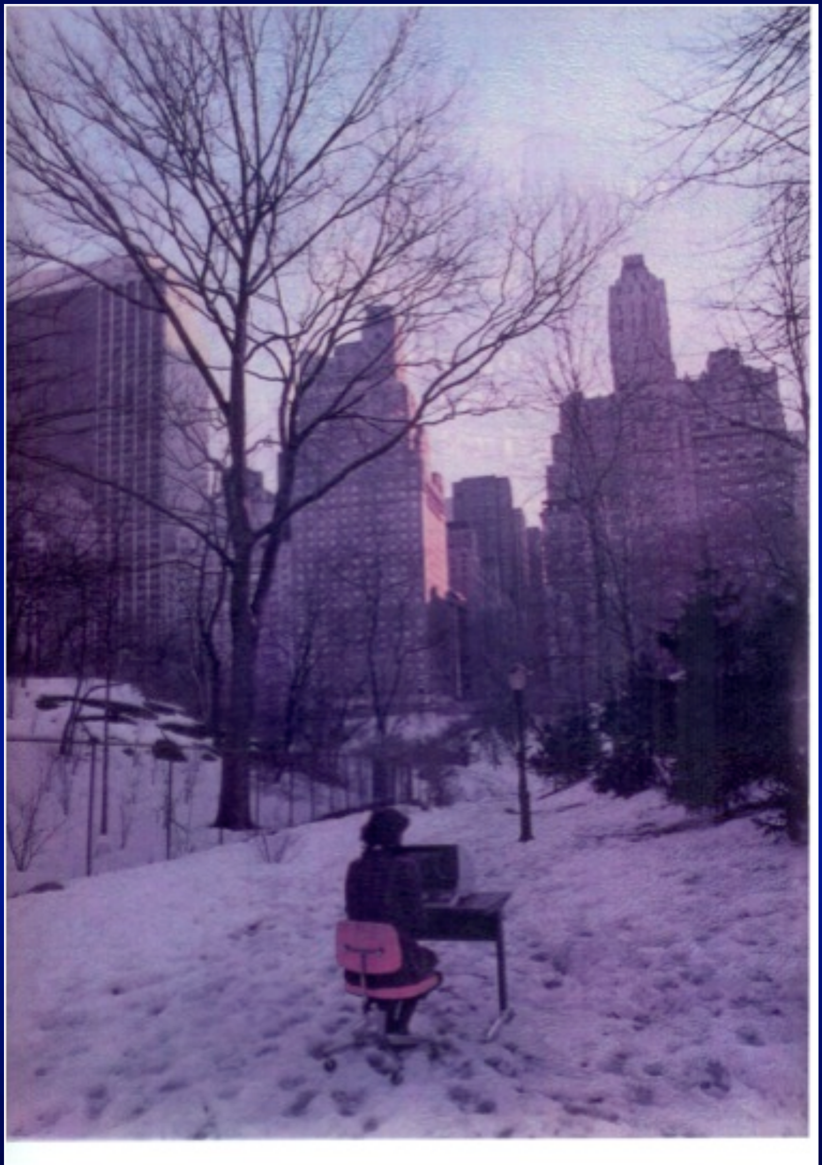
©Leonard Kleinrock 2014

6. Packet Radio

1970's

Lots of great analysis and design, but the technology would not become available for two decades more

©Leonard Kleinrock 2014



©Leonard Kleinrock 2014

Slotted Aloha

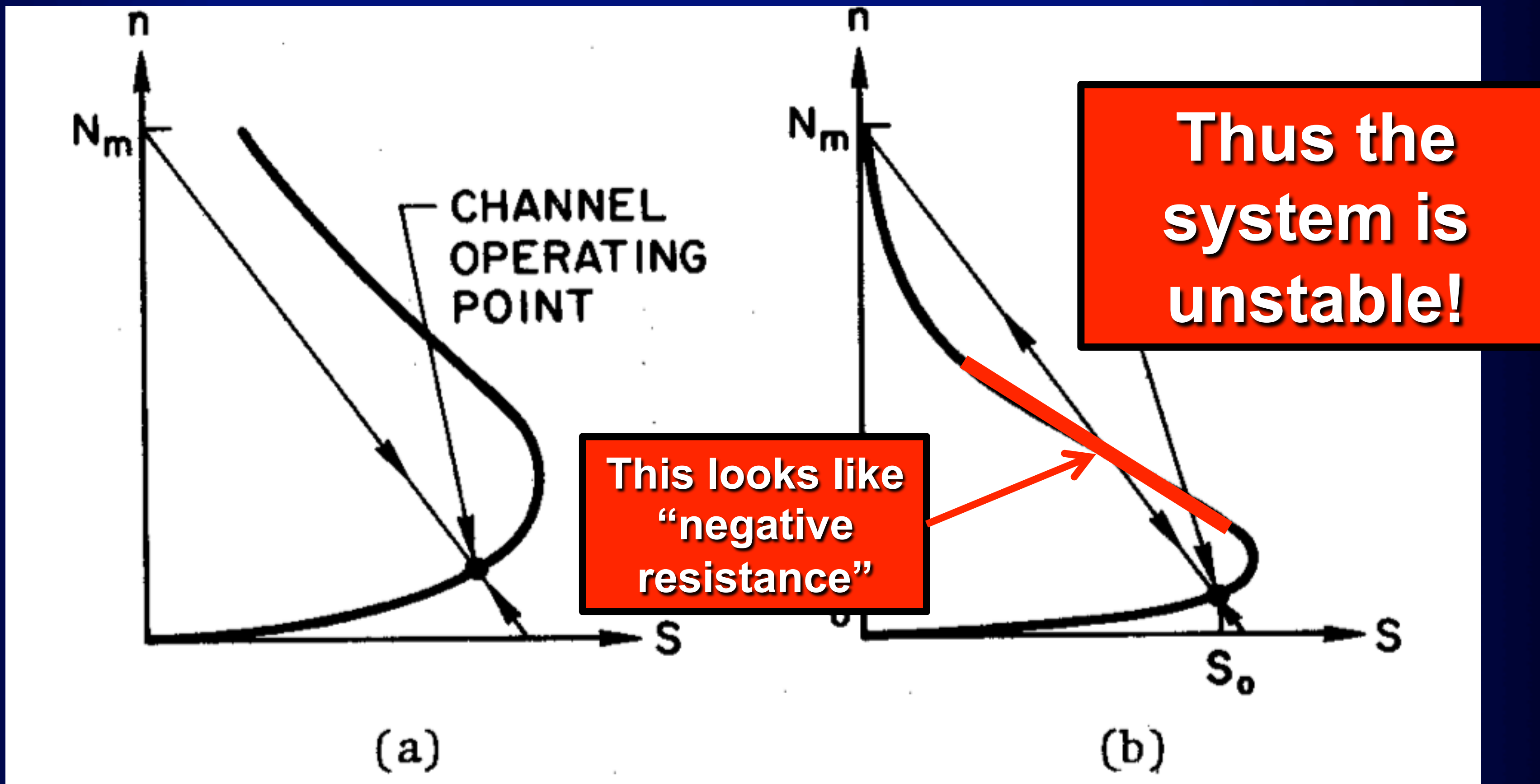


Fig. 5 Stable and Unstable Channels

L. Kleinrock and S. Lam, "Packet Switching in a Slotted Satellite Channel," in AFIPS Conference Proceedings, National Computer Conference, New York, June 1973, pp. 703–710.

©Leonard Kleinrock 2014

CSMA

1-Persistent CSMA

$$S=\frac{G[1+G+aG(1+G+aG/2)]e^{-G(1+2a)}}{G(1+2a)-(1-e^{-aG})+(1+aG)e^{-G(1+a)}} \quad (1)$$

Slotted 1-Persistent CSMA

$$S=\frac{Ge^{-G(1+a)}[1+a-e^{-aG}]}{(1+a)(1-e^{-aG})+ae^{-G(1+a)}} \quad (2)$$

Non-Persistent CSMA

$$S=\frac{Ge^{-aG}}{G(1+2a)+e^{-aG}} \quad (3)$$

Slotted Non-Persistent CSMA

$$S=\frac{aGe^{-aG}}{(1+a)(1-e^{-aG})+a} \quad (4)$$

p-Persistent CSMA

$$S(G,p,a)=\frac{(1-e^{-aG})[P_s'\pi_0+P_s(1-\pi_0)]}{(1-e^{-aG})[a\bar{t}'\pi_0+a\bar{t}(1-\pi_0)+1+a]+a\pi_0} \quad (5)$$

L. Kleinrock and F. Tobagi, "Random Access Techniques for Data Transmission over Packet Switched Radio Channels," in AFIPS Conference Proceedings, National Computer Conference, Anaheim, California, May 1975, pp. 187–201.

©Leonard Kleinrock 2011

CSMA

1-Persistent CSMA

$$S=\frac{G[1+G+aG(1+G+aG/2)]e^{-G(1+2a)}}{G(1+2a)-(1-e^{-aG})+(1+aG)e^{-G(1+a)}} \qquad (1)$$

Slotted 1-Persistent CSMA

$$S=\frac{Ge^{-G(1+a)}[1+a-e^{-aG}]}{(1+a)(1-e^{-aG})+ae^{-G(1+a)}} \qquad (2)$$

Non-Persistent CSMA

$$S=\frac{Ge^{-aG}}{G(1+2a)+e^{-aG}}$$

Slotted Non-Persistent CSMA

On the airplane home

Plus

- **Hidden Terminals**
- **Busy Tone**
- **Reservation**

L. Kleinrock and F. Tobagi, "Random Access Techniques for Data Transmission over Packet Switched Radio Channels," in AFIPS Conference Proceedings, National Computer Conference, Anaheim, California, May 1975, pp. 187–201.

©Leonard Kleinrock 2011

Distributed Multi-Access

- Performance degradation from pure queueing due to:
  - Unpredictable arrival times
  - Unpredictable service times
- We also lose performance because we do not know who is on queue in a distributed environment

©Leonard Kleinrock 2014

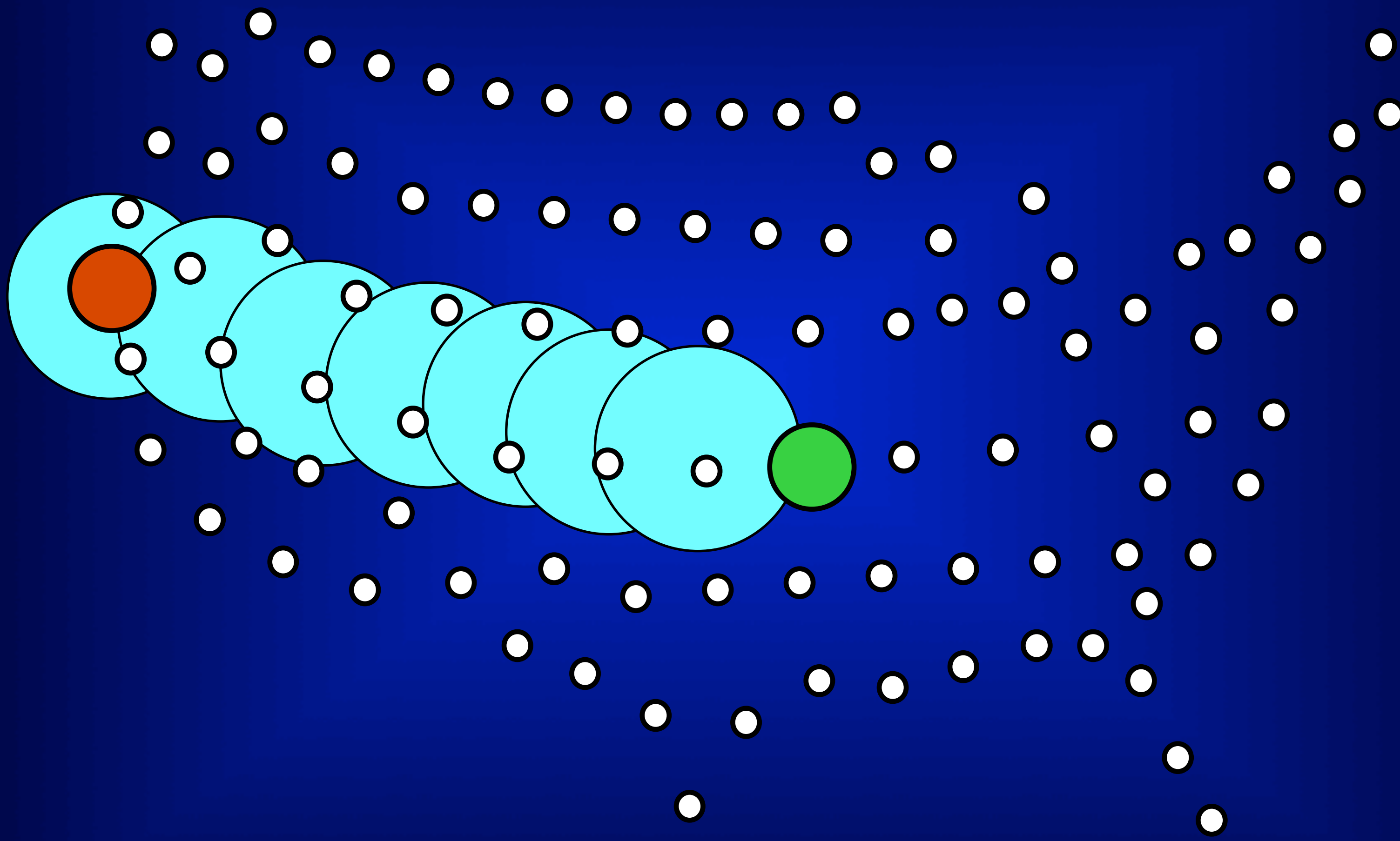
The Price for Forming the Queue

	Collisions	Idle Capacity	Control Overhead
No Control (e.g. Aloha)	Yes	No	No
Static Control (e.g. FDMA)	No	Yes	No
Dynamic Control (e.g. Reservation)	No	No	Yes

L. Kleinrock, "Performance of Distributed Multi-Access Computer-Communication Systems," in Information Processing 77, Proceedings of IFIP Congress 77, Toronto, Canada, August 1977, pp. 547–552.

©Leonard Kleinrock 2014

Giant Stepping
in Packet Radio



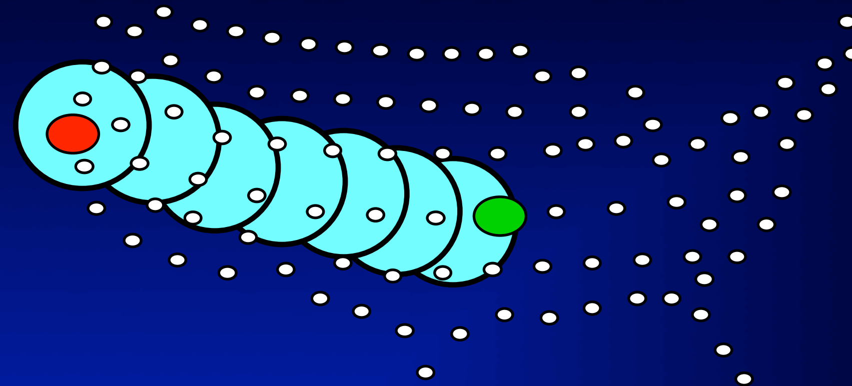
©Leonard Kleinrock 2014

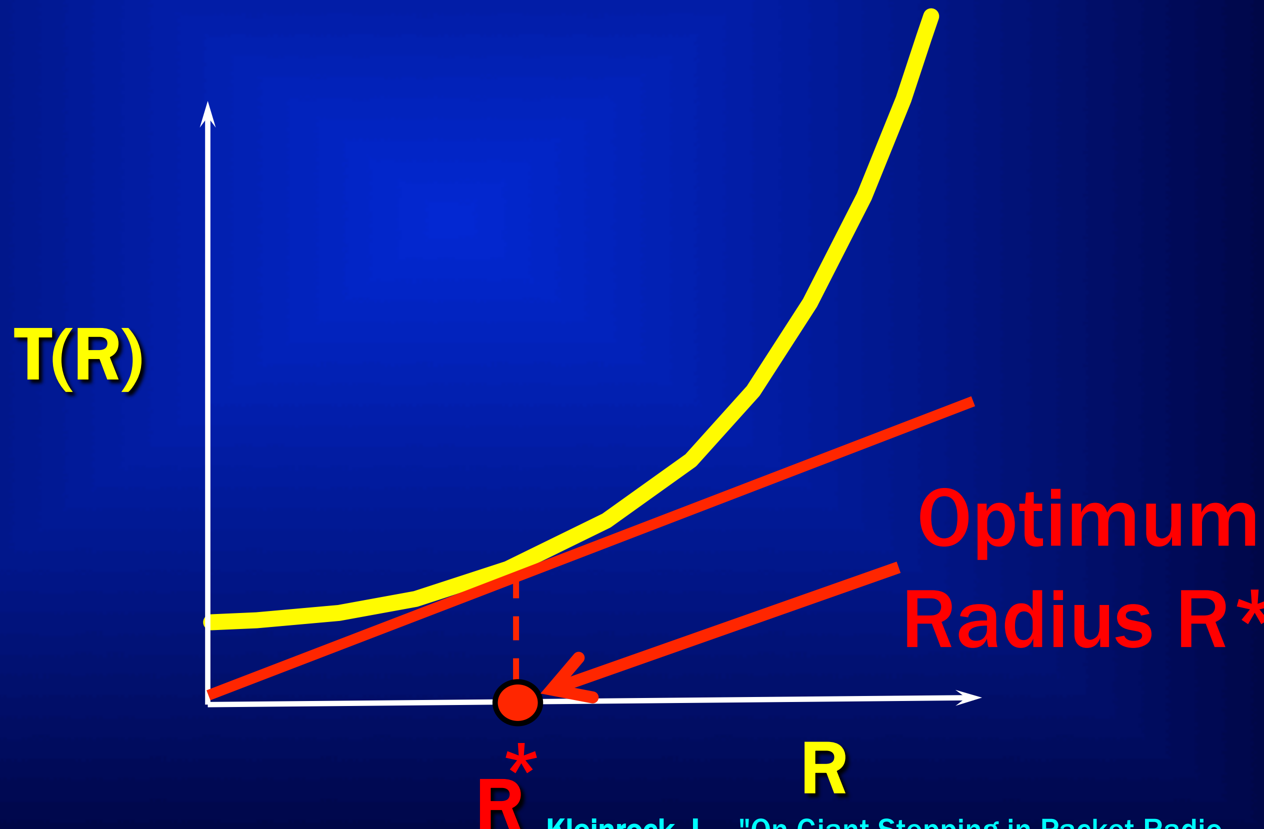
Giant Stepping
in Packet Radio

- Multihop
- Each hop covers distance R (Tx Radius)
- Total distance to cover is D (D>>R)
- Big R, more interference, fewer hops
- Small R, less interference, more hops
- T(R) is mean response time per hop
- T=Total Delay = T(R)[D/R]
- Choose $R=R^*$ to minimize total delay
- dT(R)/dR = T(R)/R optimality condition

©Leonard Kleinrock 2014

$$dT(R)/dR = T/R$$





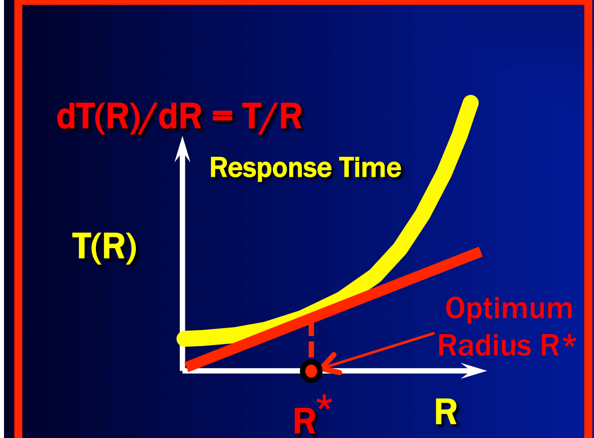
Kleinrock, L., "On Giant Stepping in Packet Radio Networks," UCLA, Packet Radio Temporary Note #5, Prt 136, March 1975

©Leonard Kleinrock 2014

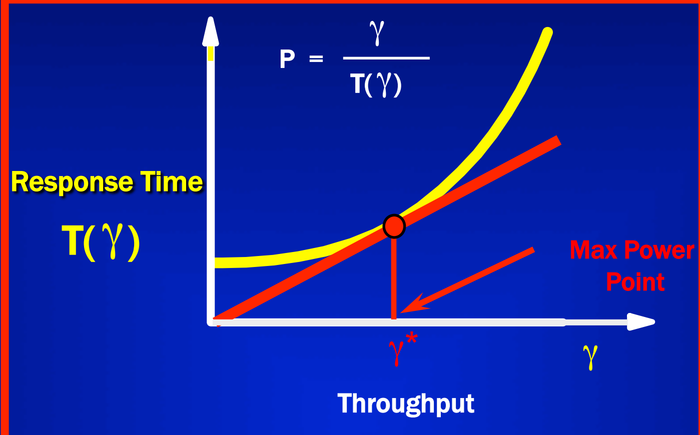
7. A Generalization

©Leonard Kleinrock 2014

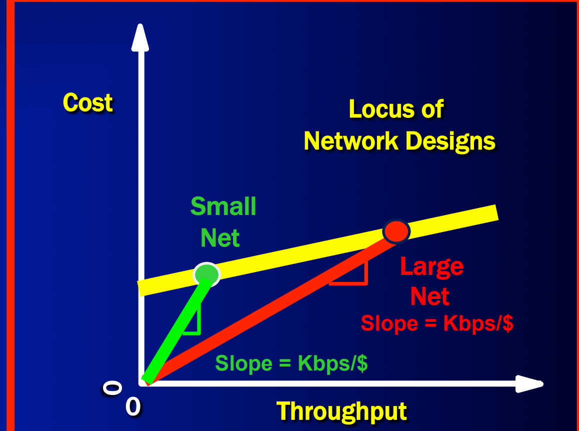
This is the 3rd Time We Have Seen This Today!



Giant Stepping



Power

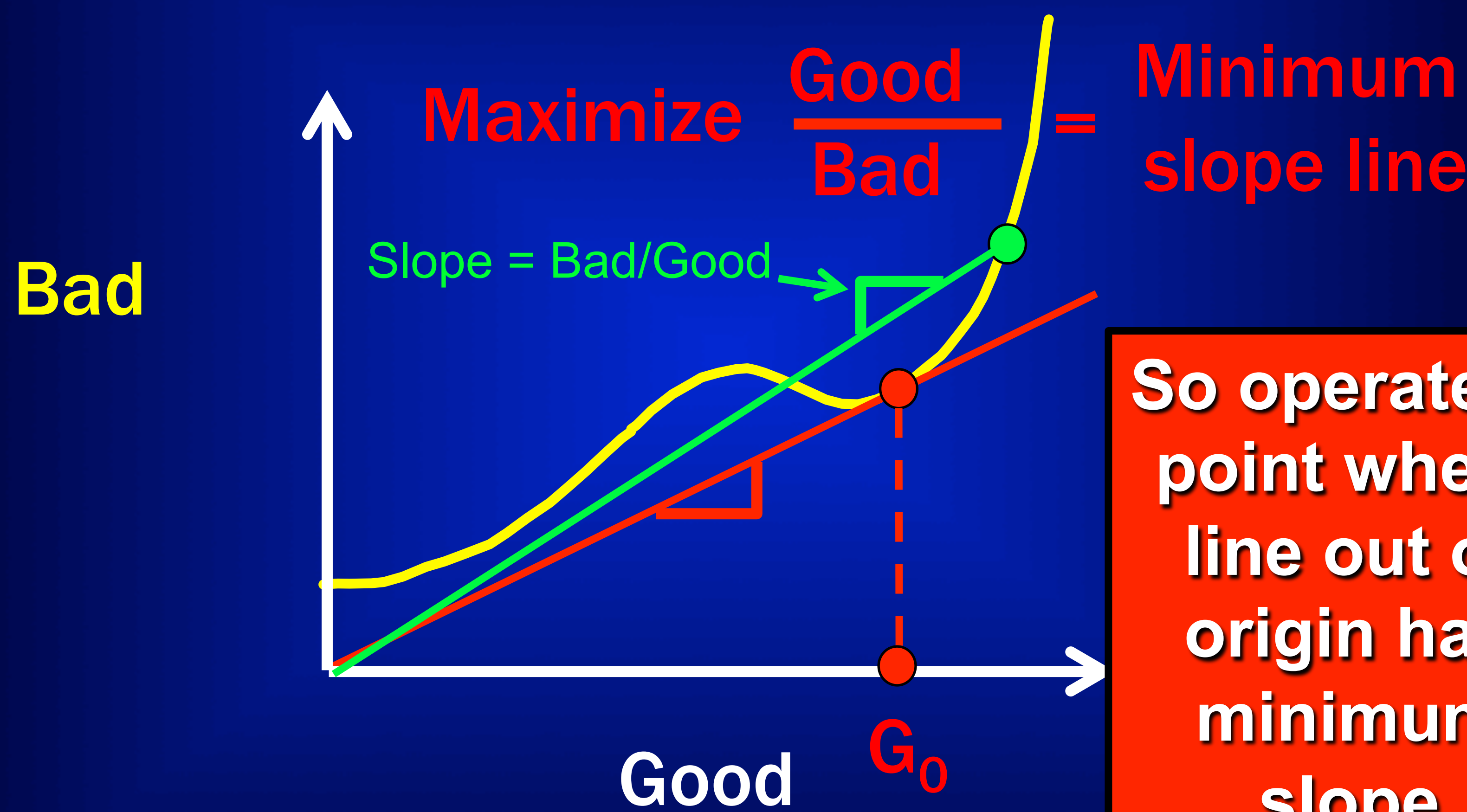


Economy of Scale

Is there a General Case Here?

©Leonard Kleinrock 2014

The General Case



So operate at point where line out of origin has minimum slope

Kleinrock, L., "Optimizing the Ratio of Good/Bad" in preparation

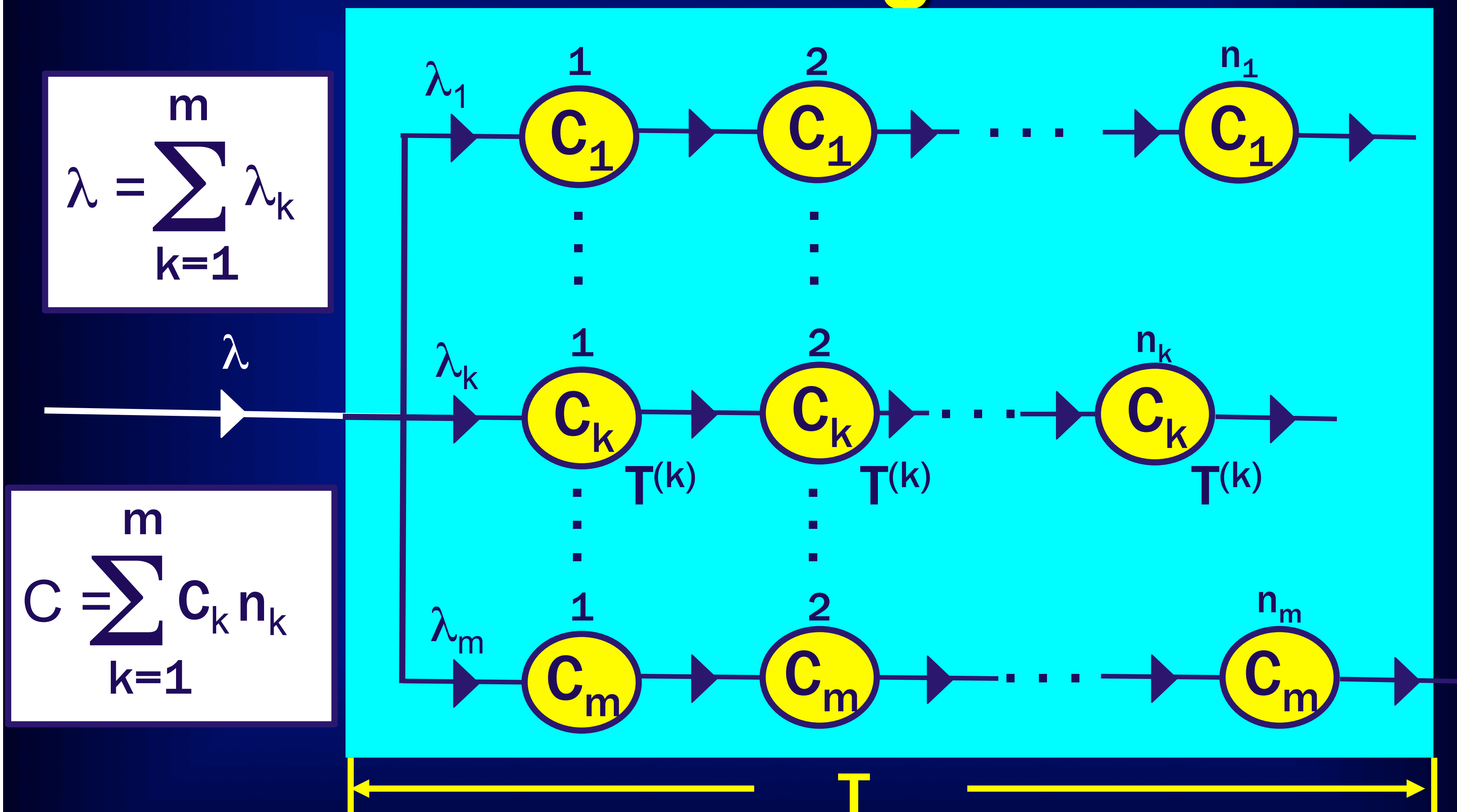
©Leonard Kleinrock 2014

8. Distributed Processing

1980's

©Leonard Kleinrock 2014

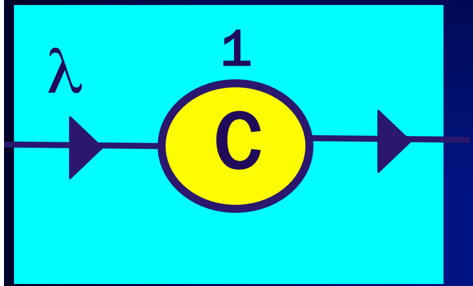
The General Series/Parallel Processing Net



Kleinrock, L., "On the Theory of Distributed Processing," *Proc of the Twenty-Second Annual Allerton Conference on Communication, Control and Computing*, Urbana-Champaign, October 1984, pp. 60–70.

©Leonard Kleinrock 2014

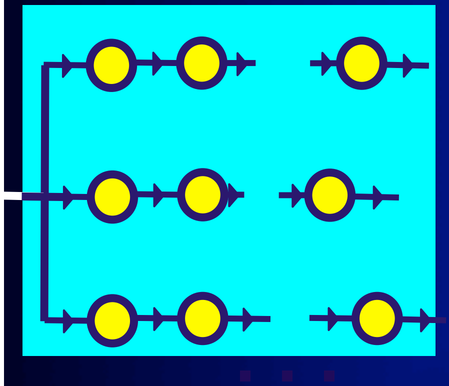
The Pure Single Node



$$T_0 = \frac{1}{\mu C - \lambda} \qquad M/M/1$$

$1/\mu$ = Avg No. of opns/job

The General Series/Parallel



$$T = \sum_{k=1}^{m} \frac{\lambda_k}{\lambda} n_k T^{(k)}$$

$$T^{(k)} = \frac{1}{\mu n_k C_k - \lambda_k}$$

©Leonard Kleinrock 2014

Ratio of General/Single Node

$$\frac{T}{T_0} = \sum_{k=1}^{m} n_k \frac{\rho_k / (1 - \rho_k)}{\rho / (1 - \rho)}$$

$$\rho_k = \lambda_k / \mu n_k C_k$$

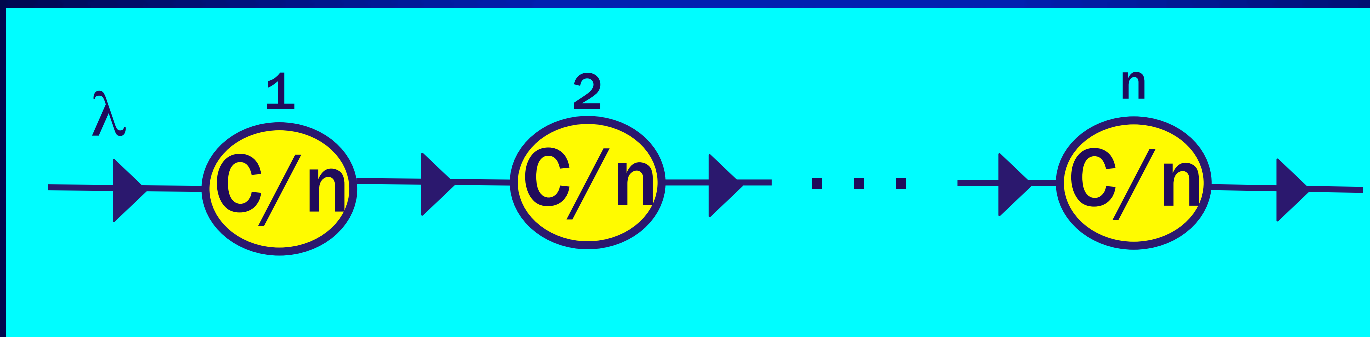
$$\rho = \lambda / \mu C$$

Let's look at some special cases:

©Leonard Kleinrock 2014

The Pure Tandem

- m=1, n_1=n, $\lambda_1 = \lambda$, C_1 = C/n

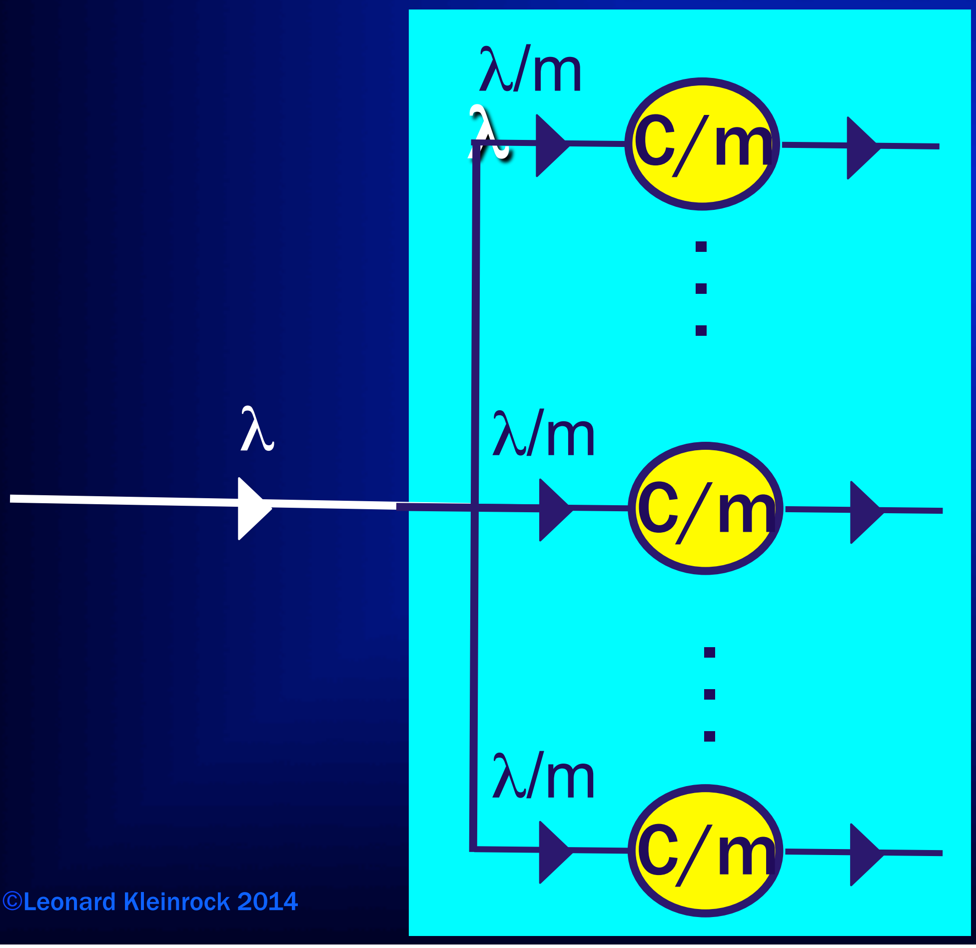


$$\frac{T}{T_0} = n$$

©Leonard Kleinrock 2014

The Pure Parallel System

- $n_k = 1$, $\lambda_k = \lambda/m$, $C_k = C/m$ for $k=1,2,\ldots,m$

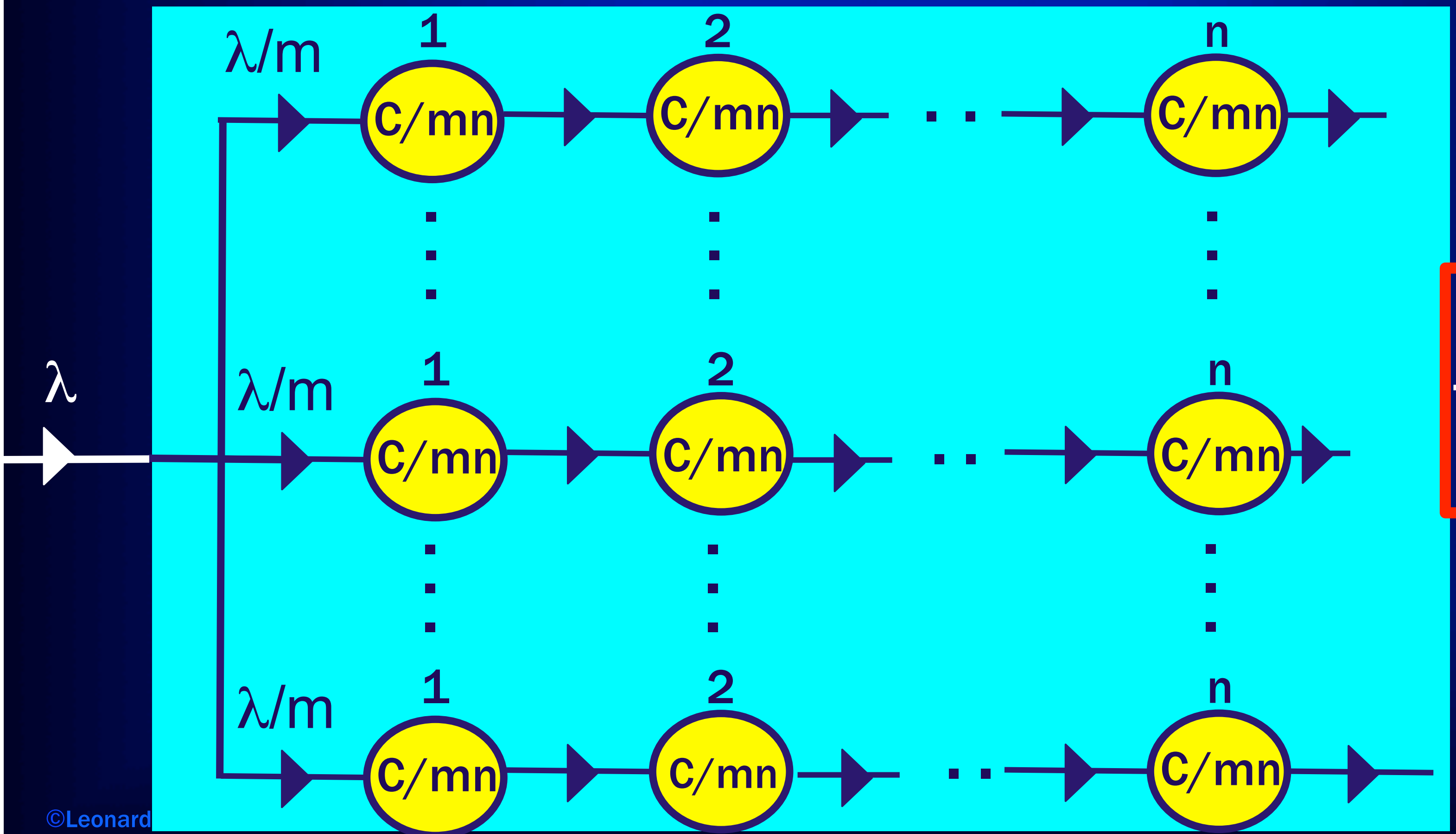


$$\frac{T}{T_0} = m$$

©Leonard Kleinrock 2014

The Symmetric Series-Parallel System

- $n_k = n, \quad \lambda_k = \lambda/m\,, \quad C_k = C/mn \quad$ for $k=1,2,\ldots,m$

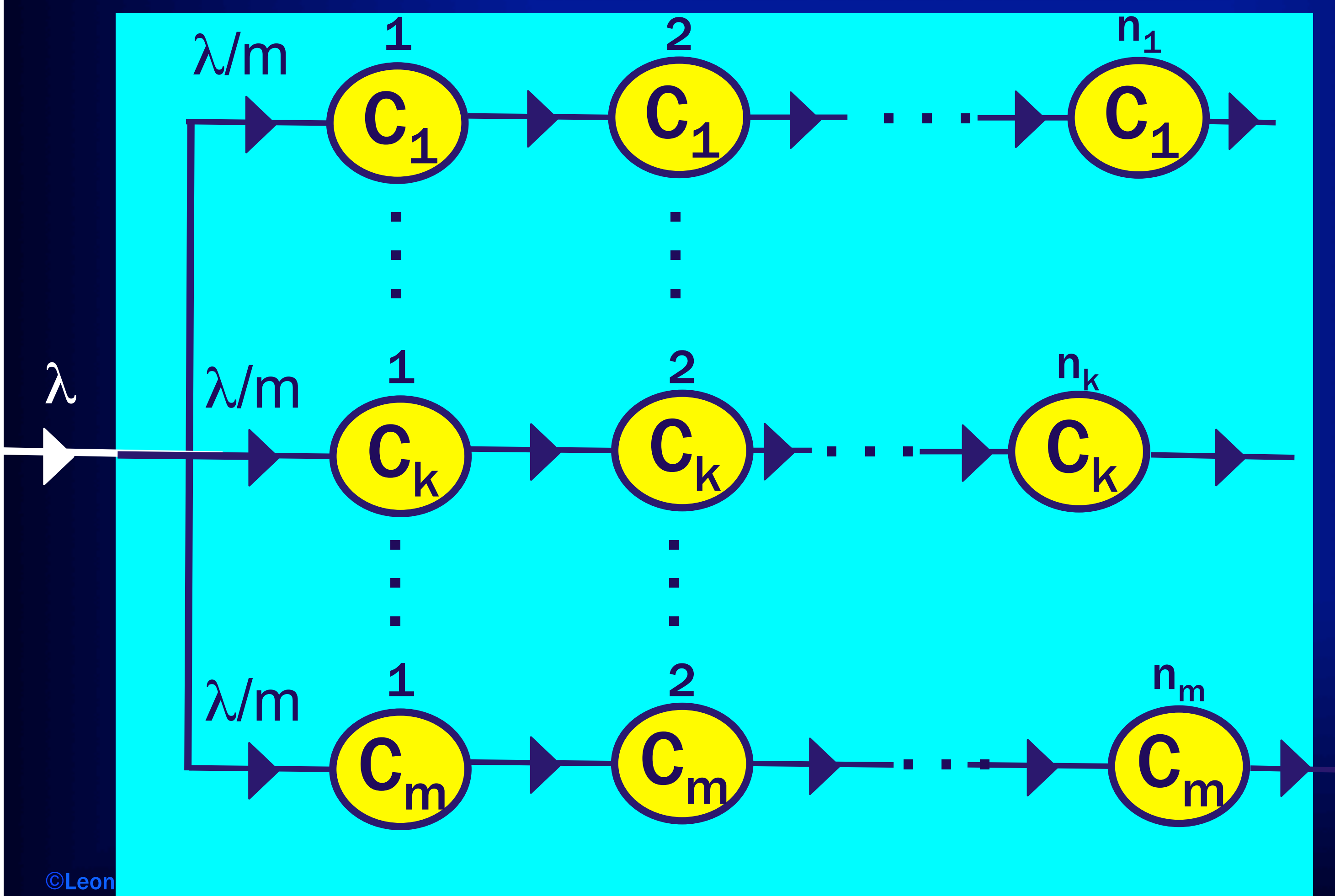


$$\frac{T}{T_0} = mn$$

©Leonard

The General Series/Parallel System with Uniform Traffic

$$\lambda_k = \lambda/m$$



$$\frac{T}{T_0} = \sum_{k=1}^{m} n_k$$

Bigger and fewer is better

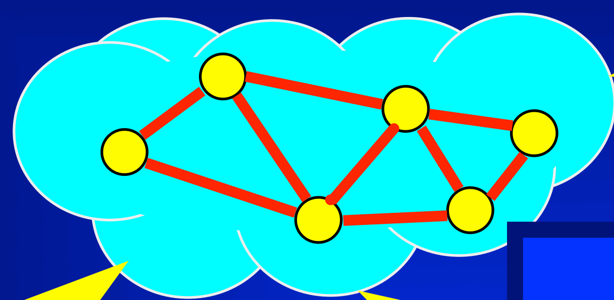
©Leon

9. Latency/Bandwidth Tradeoff

1990's

©Leonard Kleinrock 2014

The Latency/Bandwidth Tradeoff



from Kilobits

to Megabits

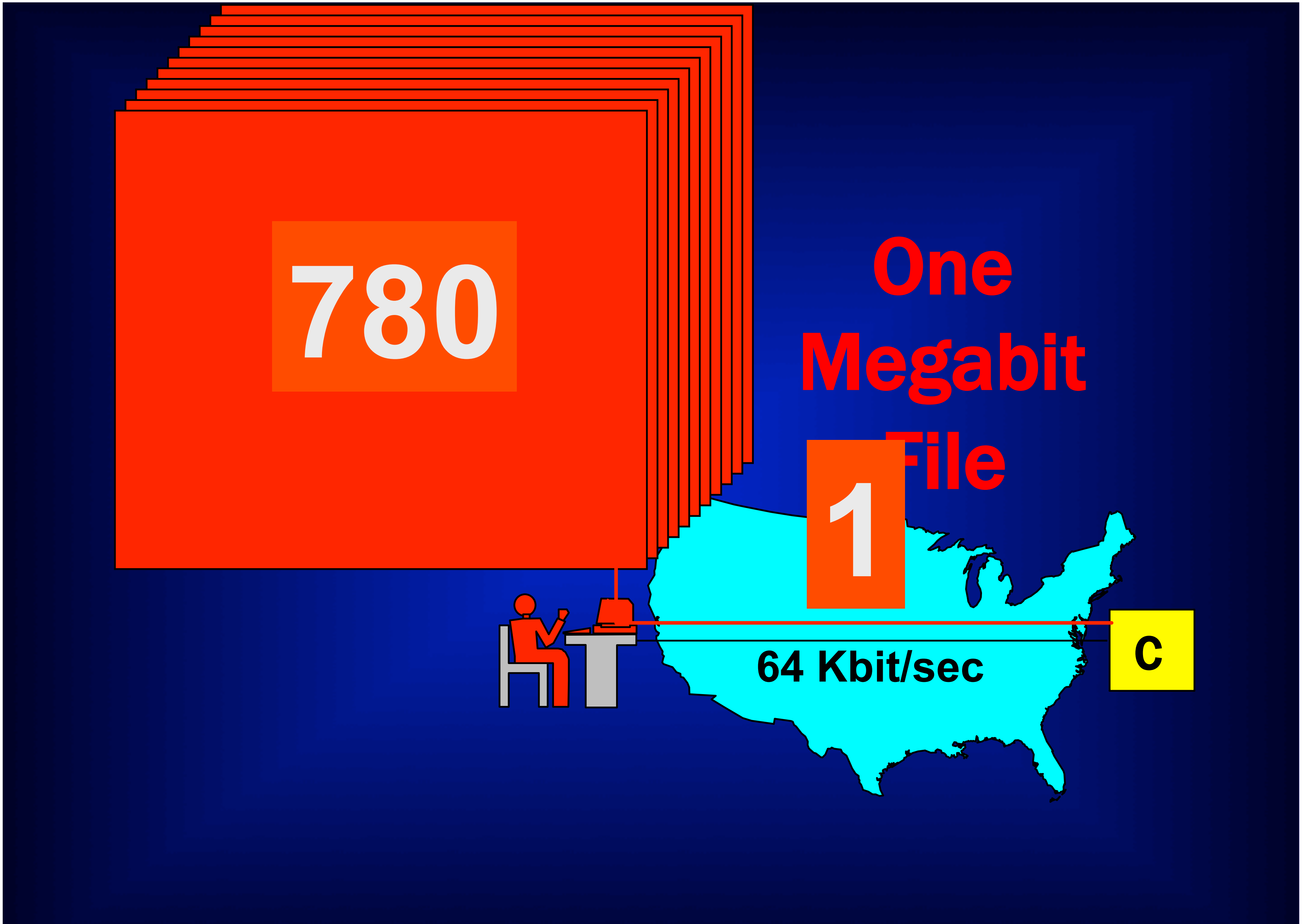
to Gigabits!

Evolution, Revolution or Bump?

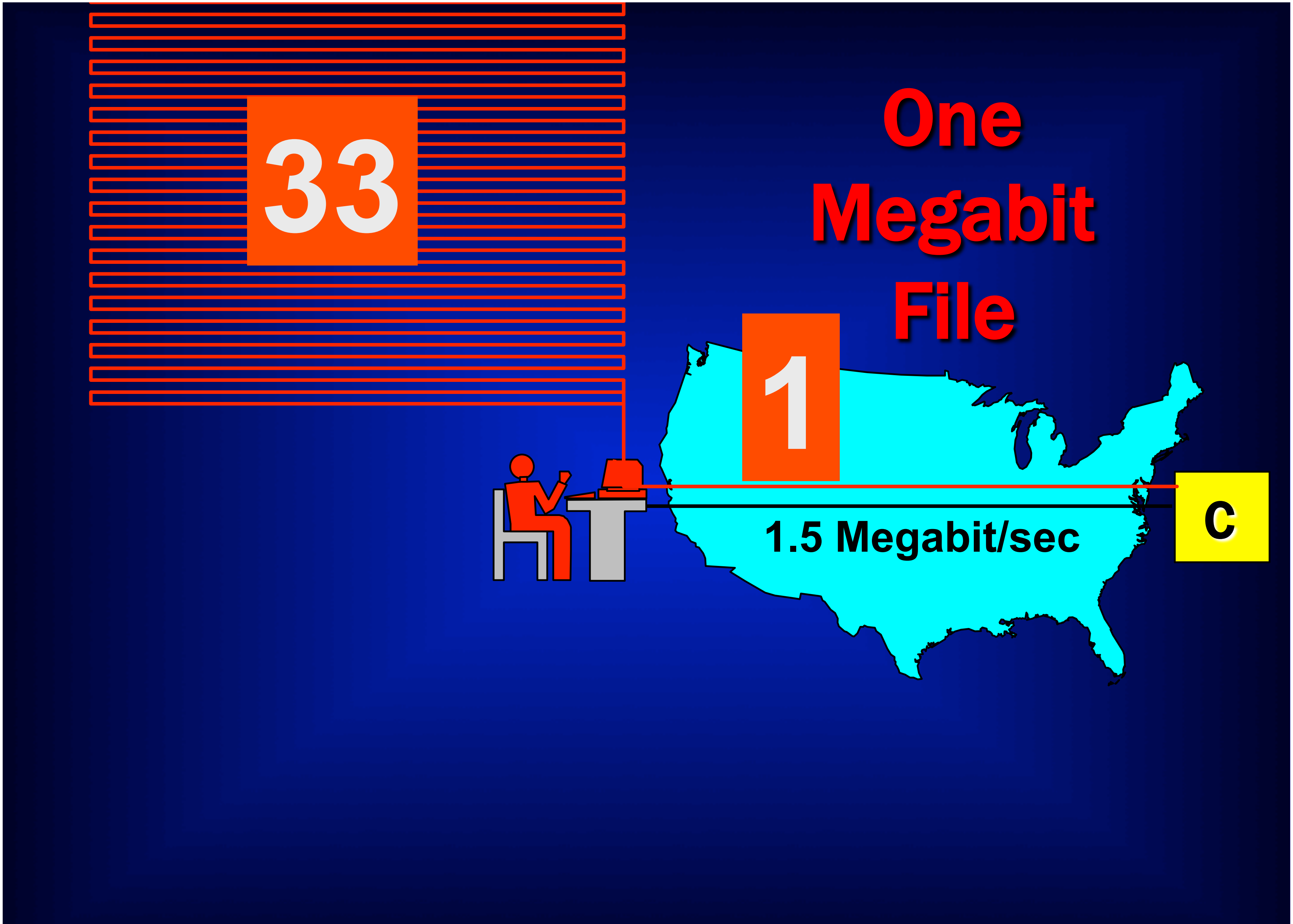
How Fast is a Gigabit?

- A billion bits/sec is really fast!
- But ... the speed of light isn't!

©Leonard Kleinrock 2014



©Leonard Kleinrock 2014



©Leonard Kleinrock 2014

We seem to have bumped into the speed of light!

or

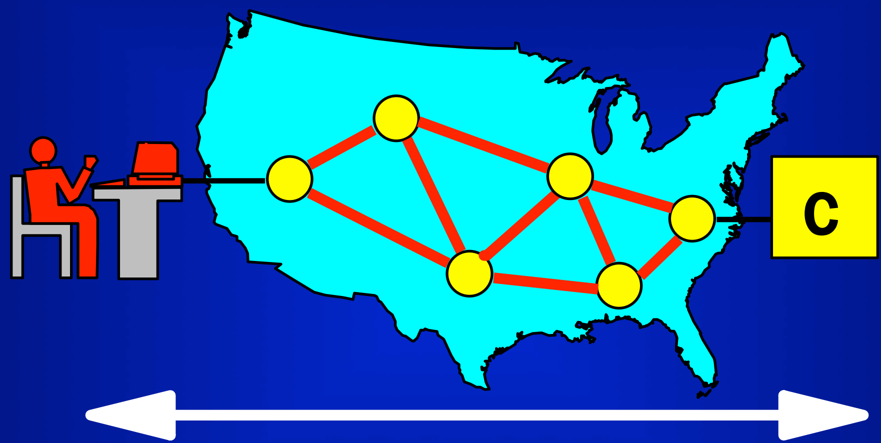
Something's going "bump" in the light!

One Megabit File

©Leonard Kleinrock 2014

When Did We Hit the Bump?

At some CRITICAL capacity!



T = Response Time

Response Time = Queueing + Tx Time + Latency

Define Critical Capacity to be the point where:

Queueing + Tx Time = Latency

©Leonard Kleinrock 2014

Kleinrock, L., "The Latency/Bandwidth Tradeoff In Gigabit Networks", *IEEE Communications Magazine*, April 1992, Vol.30, No.4, pp.36-40

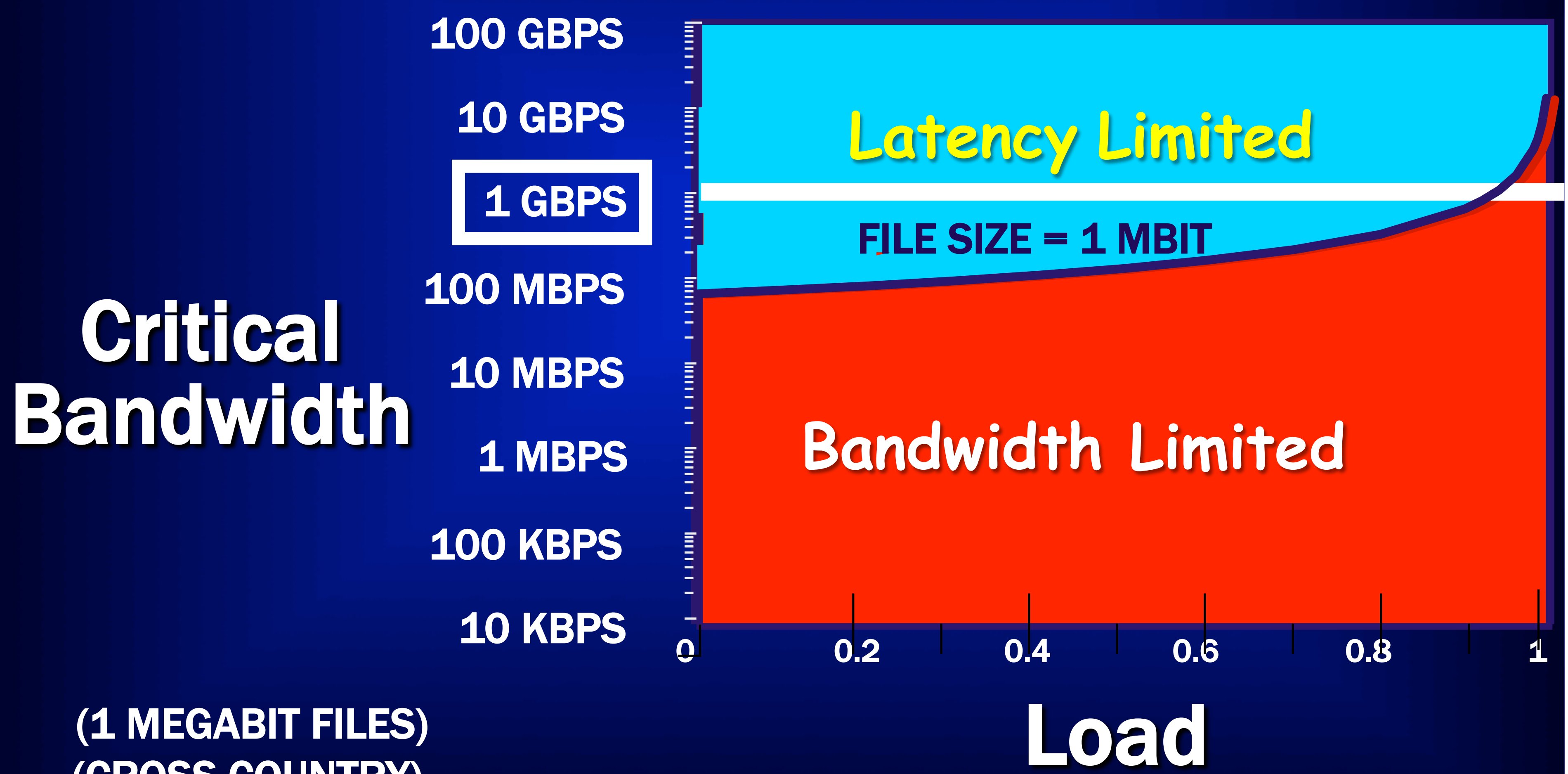
The Latency-Bandwidth Tradeoff

- Queueing + Tx Time = Latency
- C < Critical → Bandwidth Limited
- C > Critical → Latency Limited

©Leonard Kleinrock 2014

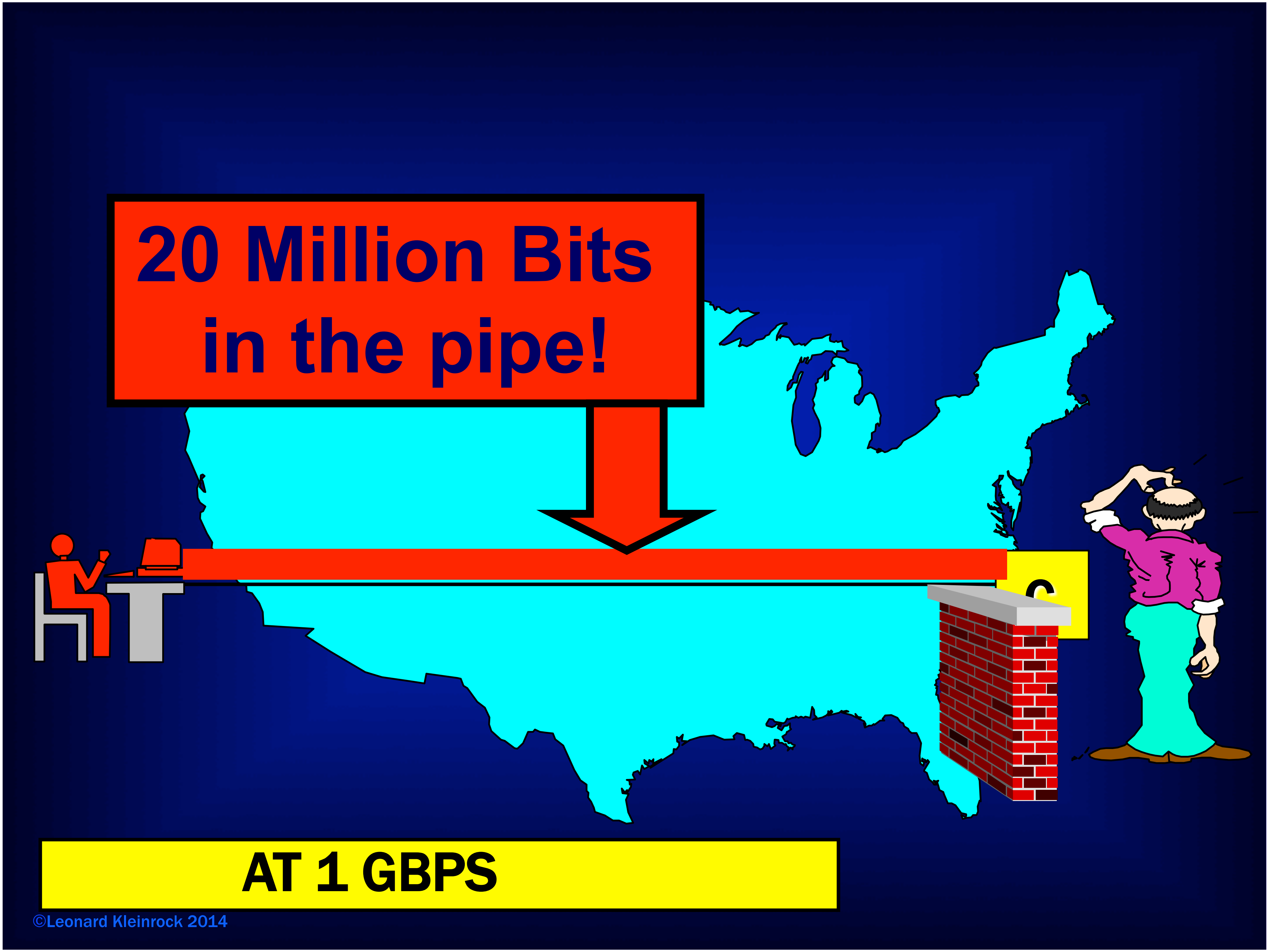
Critical Bandwidth

Queueing + Tx Time = Latency



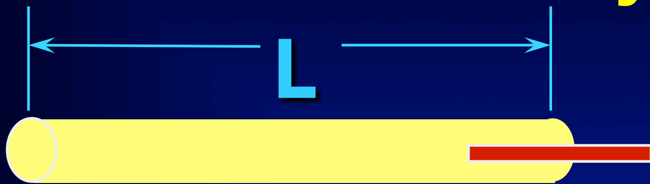
(1 MEGABIT FILES)
(CROSS COUNTRY)

©Leonard Kleinrock 2014



©Leonard Kleinrock 2014

Key System Parameter



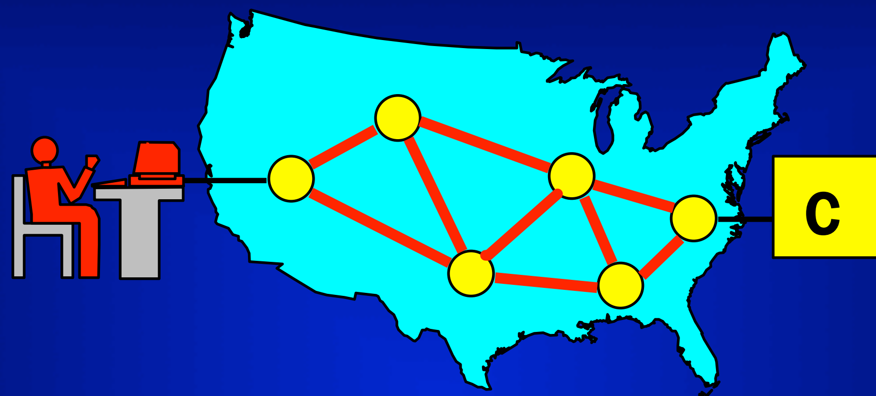
L = Cable Length (kilometers)
PD = 5L (microseconds)
C = Bandwidth (megabits/sec)
b = Packet Length (bits)

a = Propag Delay/Pkt Tx Time
= 5LC/b (# packets in cable)

	SPEED MBPS	PKT LNGTH BITS	PROP DELAY MICROSEC	LATENCY a
WIRELESS NET 1 kilometer	10.0	1,000	5	.05
LOCAL NET 1 kilometer	1,000.00	1,000	5	5
FIBER LINK Cross country	1,000.00	1,000	20,000	20,000

©Leonard Kleinrock 2014

The Latency-Bandwidth Tradeoff



$$C_{crit} = \frac{b}{5L(1-\rho)}$$

or

$$a = \frac{1}{1-\rho}$$

where

ρ = Load = $\lambda b / C$

C (Mbps), b (bits/msg)

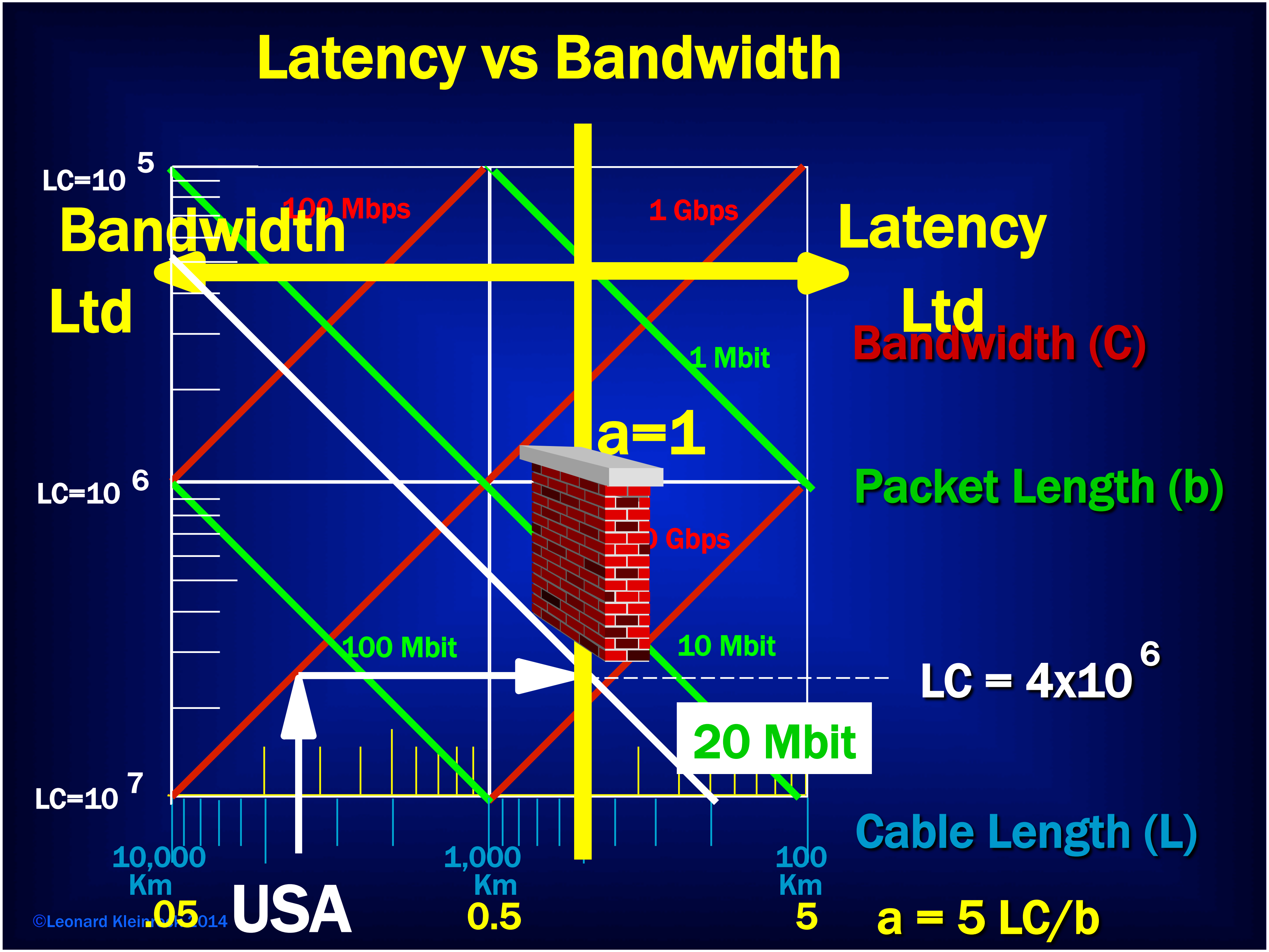
L (Km), λ(msg/microsec)

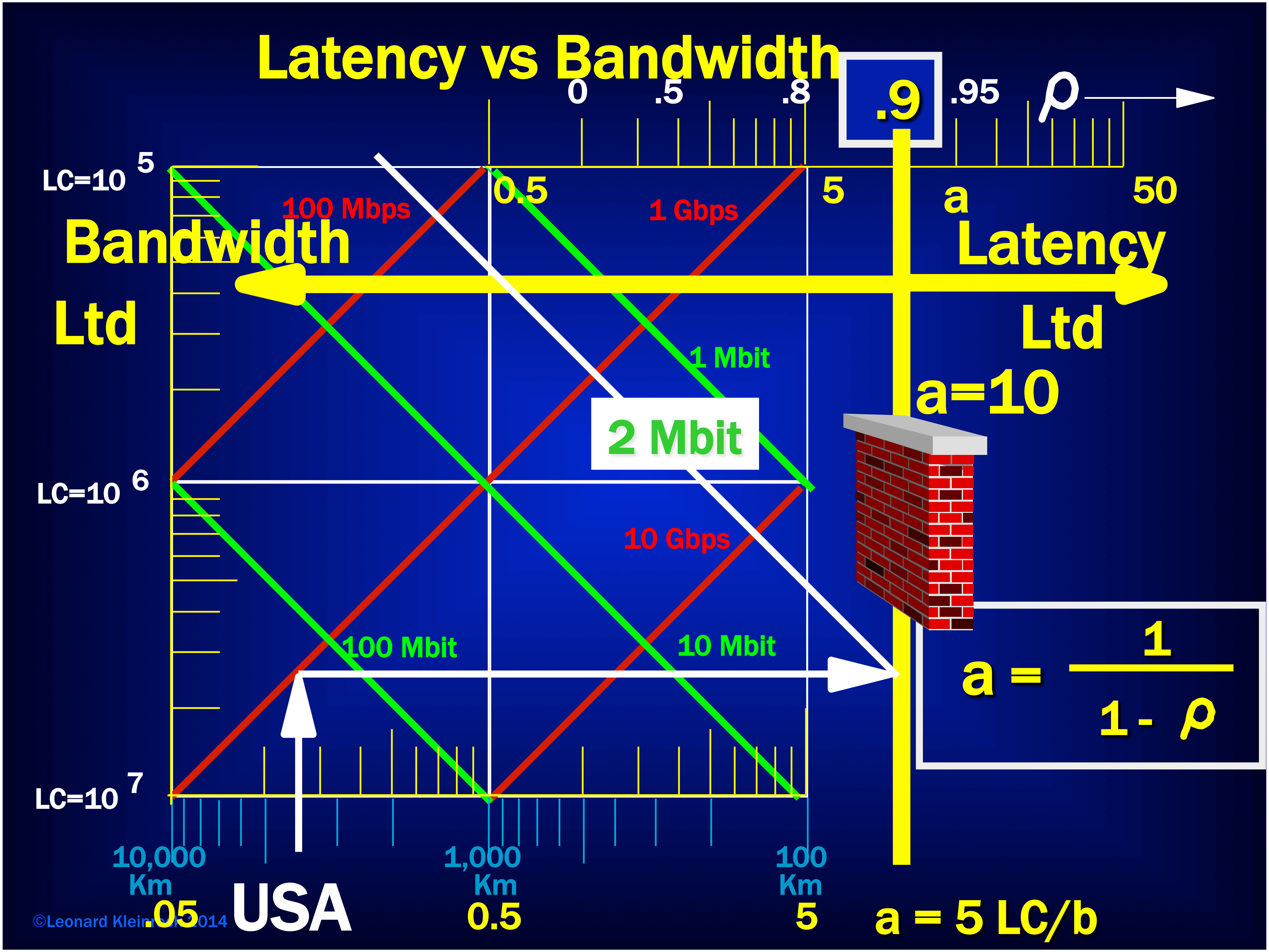
a = Propag Delay/Pkt Tx Time

= 5LC/b (# packets in cable)

©Leonard Kleinrock 2014

Kleinrock, L., "The Latency/Bandwidth Tradeoff In Gigabit Networks", *IEEE Communications Magazine*, April 1992, Vol.30, No.4, pp.36-40


Latency vs Bandwidth
LC=10 5
LC=10 6
LC=10 7
Bandwidth
Ltd
Latency
Ltd
100 Mbps
1 Gbps
1 Mbit
a=1
Bandwidth (C)
Packet Length (b)
Gbps
100 Mbit
10 Mbit
LC = 4x10 6
20 Mbit
Cable Length (L)
10,000
Km
1,000
Km
100
Km
.05
USA
0.5
5
a = 5 LC/b
©Leonard Klein 2014


Latency vs Bandwidth
0
.5
.8
.9
.95
ρ
0.5
5
a
50
LC=10 5
100 Mbps
1 Gbps
Bandwidth
Ltd
Latency
Ltd
1 Mbit
a=10
2 Mbit
LC=10 6
10 Gbps
100 Mbit
10 Mbit
a = 1 / (1 - ρ)
LC=10 7
10,000 Km
1,000 Km
100 Km
.05
USA
0.5
5
a = 5 LC/b
©Leonard Kleinrock 2014

Gigabit Networking

Fundamental Issues

- **Speed of Light is Too Slow:**
  - 20,000 Microsec to cross USA
  - 20 Million bits in a Gigabit pipe
  - Control signals suffer enormous delays
- **Global Information is Costly:**
  - It takes:
    bandwidth, time, processing, storage.
  - It will be:
    delayed, stale, wrong, incomplete.

©Leonard Kleinrock 2014

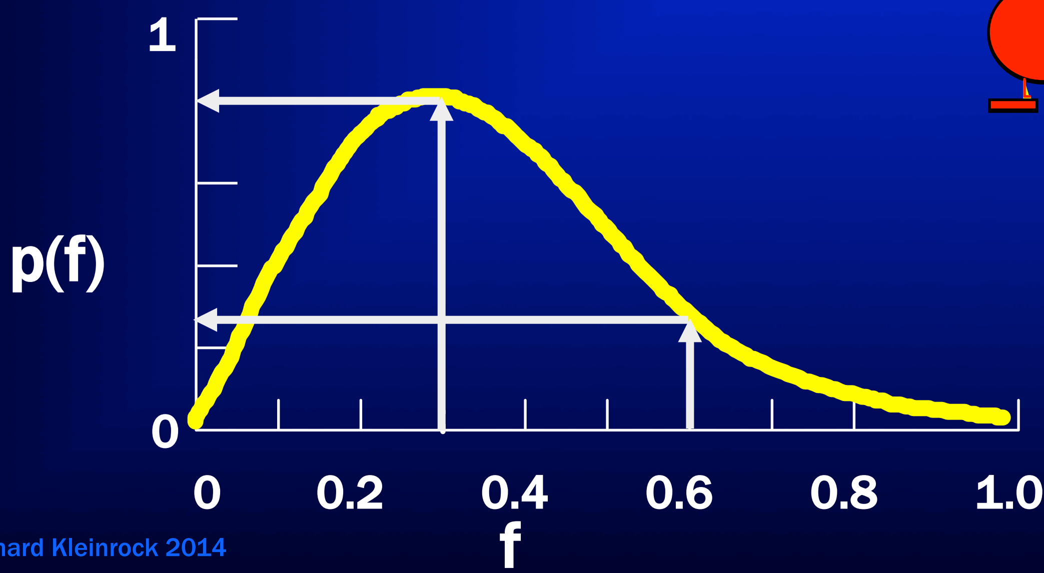
10. The Gur Intelligent Agent

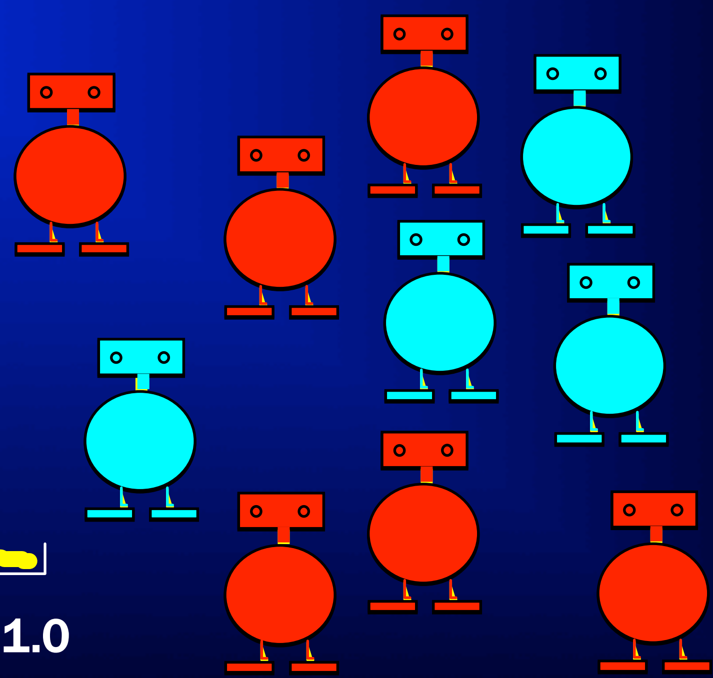
1990's

©Leonard Kleinrock 2014

Adaptive Agents and The Gur Algorithm

1. Each Agent votes YES or NO
2. A fraction f votes YES
3. Using a function p(f) which is unknown to them, a referee gives (takes) $1 from each independently with probability p
4. Go to step 1 and repeat!





©Leonard Kleinrock 2014

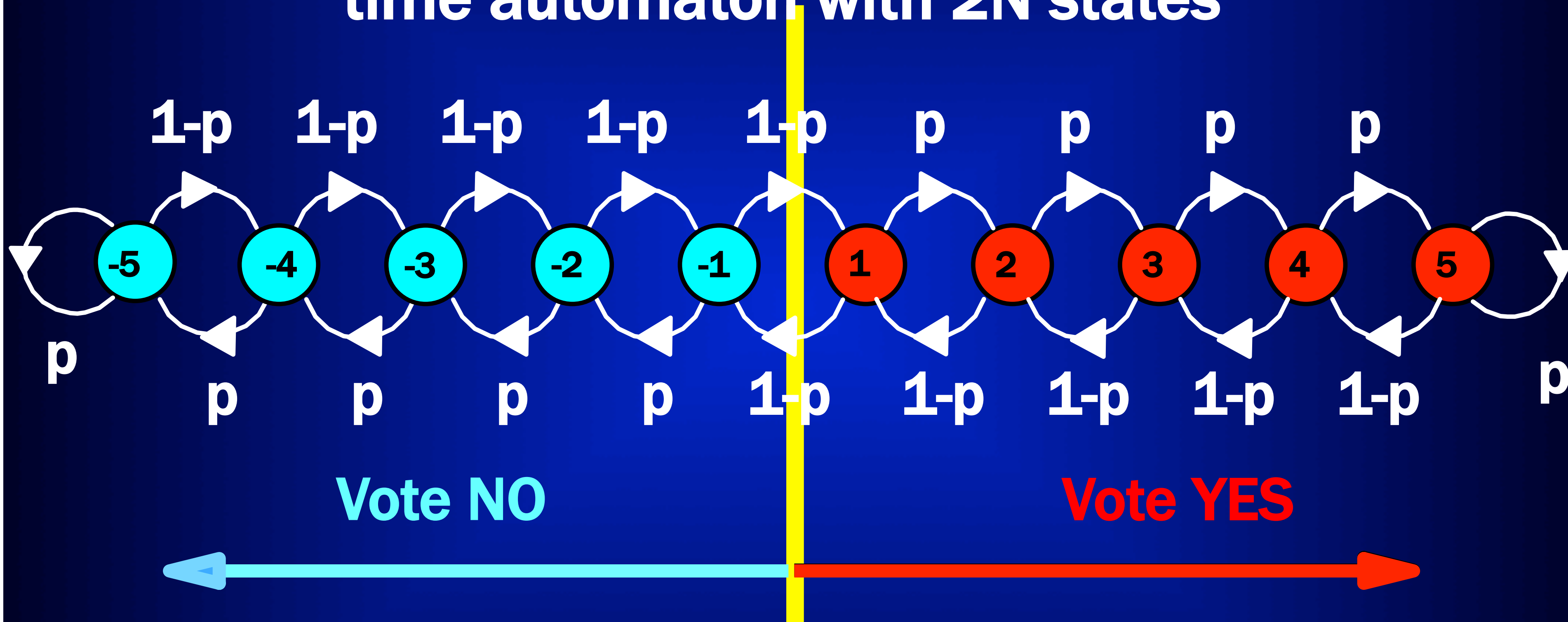
Can We Construct The Players to Seek the Optimum Behavior?

Yes !

©Leonard Kleinrock 2014

How Is It Done?

Design each player as a finite-state discrete-time automaton with 2N states



Reward => Edge seeking behavior
Punishment => Center seeking behavior

B. Tung and L. Kleinrock, "Distributed Control Methods," in Proceedings of the 2nd International Symposium on High Performance Distributed Computing, Spokane, Washington, July 21-23, 1993, pp. 206–215.

©Leonard Kleinrock 2014

11. Optimal Update Times

2000’s

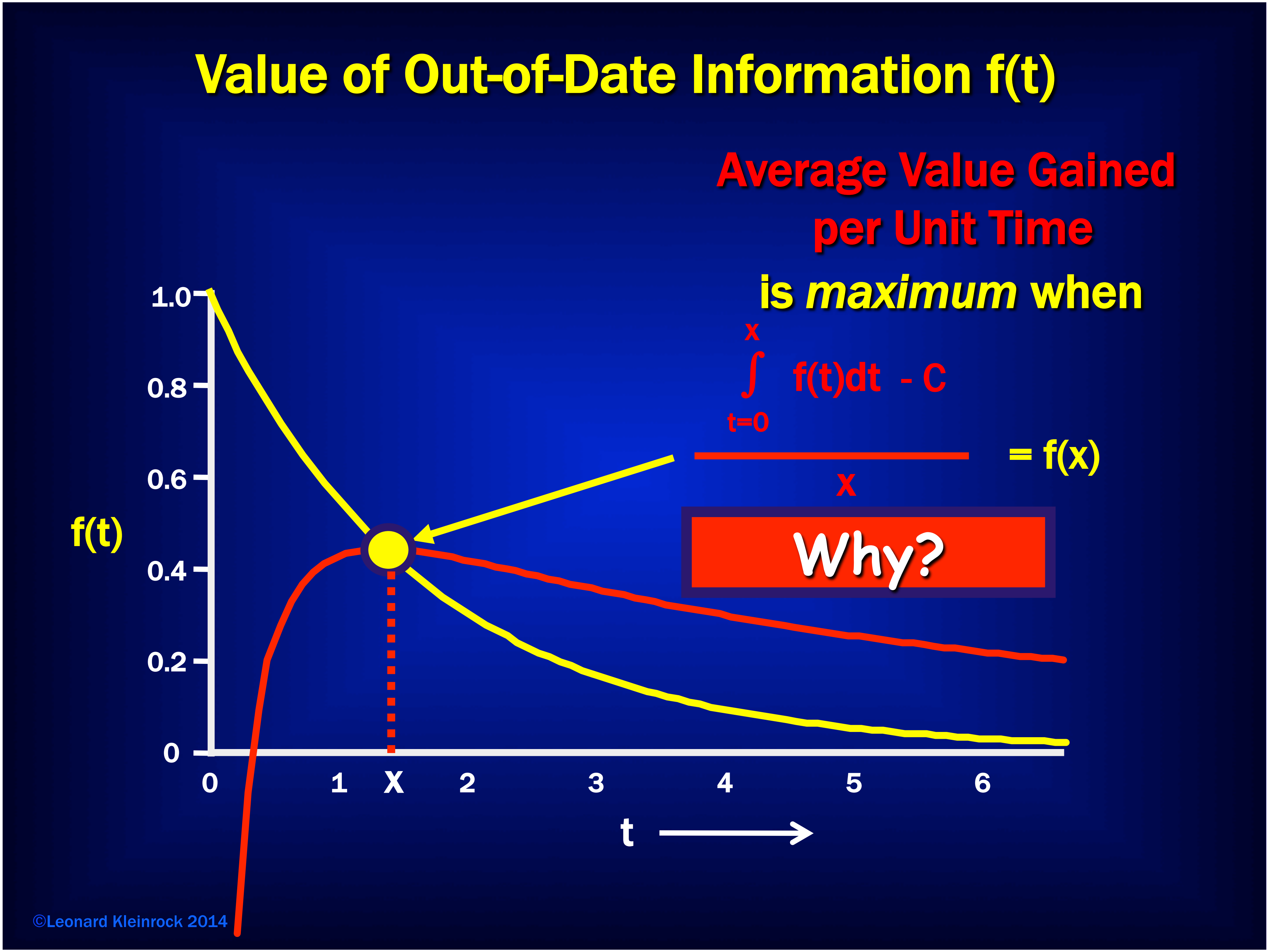
©Leonard Kleinrock 2014

Optimal Update Times for Out-of-Date Information

- **Problem:**
  - When and how often should a user update a given piece of information as it goes further and further out-of-date?
- **Assumptions:**
  - There is a cost $C>0$ of updating a given piece of information
  - There is an expected value per unit time associated with having a piece of information that was updated t time units ago.
    - This value is $f(t)$.
- **Question:**
  - Given $f(t)$ and C, When and how often should a user update a given piece of information?

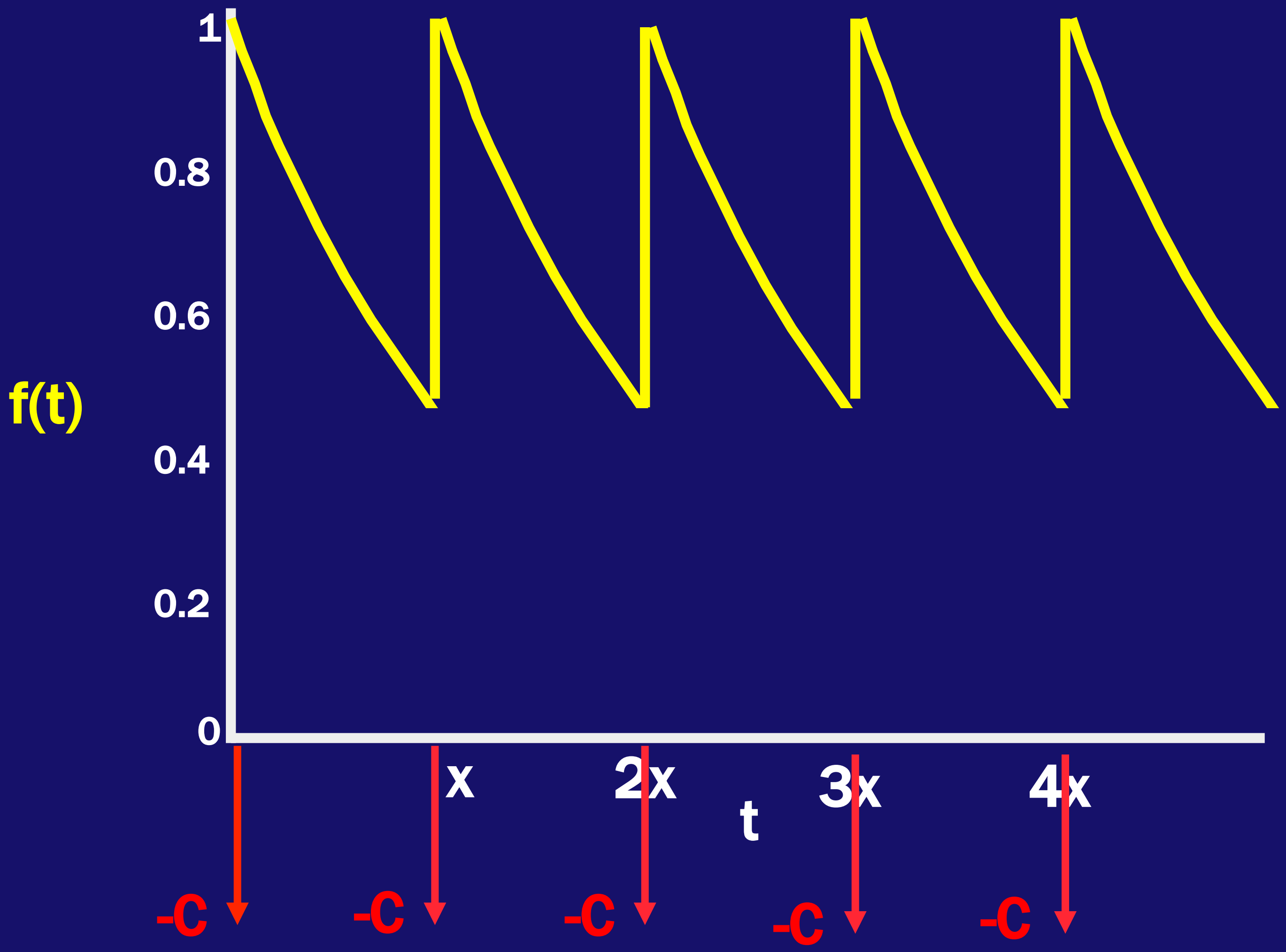
©Leonard Kleinrock 2014

Ferguson, C. and Kleinrock, L., "Optimal Update Times for Out-of-Date Information," in preparation



Value of Out-of-Date Information f(t)
Average Value Gained
per Unit Time
is maximum when
x
∫ f(t)dt - C
t=0
x
= f(x)
Why?
1.0
0.8
0.6
0.4
0.2
0
f(t)
0
1
X
2
3
4
5
6
t
©Leonard Kleinrock 2014

Value Gained Over Multiple Updates



12. Peer-to-Peer File Systems

2000's

©Leonard Kleinrock 2014

Peer-to-Peer File Networks

- Distributed file sharing network
- The service **consumers** are the service **providers** as well
- Files uniformly distributed in net
- Search using controlled flooding
- **How many copies of a file should be stored?**

©Leonard Kleinrock 2014

Definitions

- M = number of nodes in the system
- N = number of unique files in the system
- K = per-node storage size in number of files
- λ_i = request rate for file i per node
- $\lambda = \sum_{i=1}^{N} \lambda_i$ = total input rate per node
- n_i = number of replicas of file i in the system
- **How should select n_i ?**

©Leonard Kleinrock 2014

Minimum Search Distance

$\tau_i(n_i)$ = Average shortest distance from a querying node to a replica of file *i*

$$\tau_i(n_i) = \alpha \log \frac{M}{n_i}$$

$$\tau = \text{Avg search distance} = \sum_{i=1}^{N} \frac{\lambda_i}{\lambda} \tau_i(n_i)$$

$$\text{Minimize}_{\{n_i\}} \ \tau$$

$$n_i = \lambda_i \frac{KM}{\lambda}$$

©Leonard Kleinrock 2014

Further Results

Why shouldn't I store only unpopular files?

Given $$n_i = \lambda_i \frac{KM}{\lambda}$$

- Each replica of file i serves

$$M \lambda_i / n_i = \frac{\lambda}{K} \text{ requests/sec}$$

- Each node has K files, so the load on each node is λ requests/sec . **Don't play games.**
- **So each node has exactly the same load!**
- If queueing delays are convex in node utilization, **the average download time is minimized.**

S. Tewari and **L. Kleinrock**, "On Fairness, Optimal Download Performance and Proportional Replication in Peer-to-Peer Networks," in Proceedings of IFIP Networking 2005, Waterloo, Canada, May 2005.

©Leonard Kleinrock 2014

13. Guidelines for Research

©Leonard Kleinrock 2014

My Five Golden Guidelines to Research

1. Conduct the 100-year test.
2. Don't fall in love with your model.
3. Beware of mindless simulation.
4. Understand your own results.
5. Look for "Gee, that's funny!"

©Leonard Kleinrock 2014

Richard Hamming

"Why do so few scientists make significant contributions and so many are forgotten in the long run?"

"If you don't work on important problems, it's not likely that you'll do important work."

Richard W. Hamming, "You and Your Research", March 7, 1986.

©Leonard Kleinrock 2014

1. The 100 Year Test

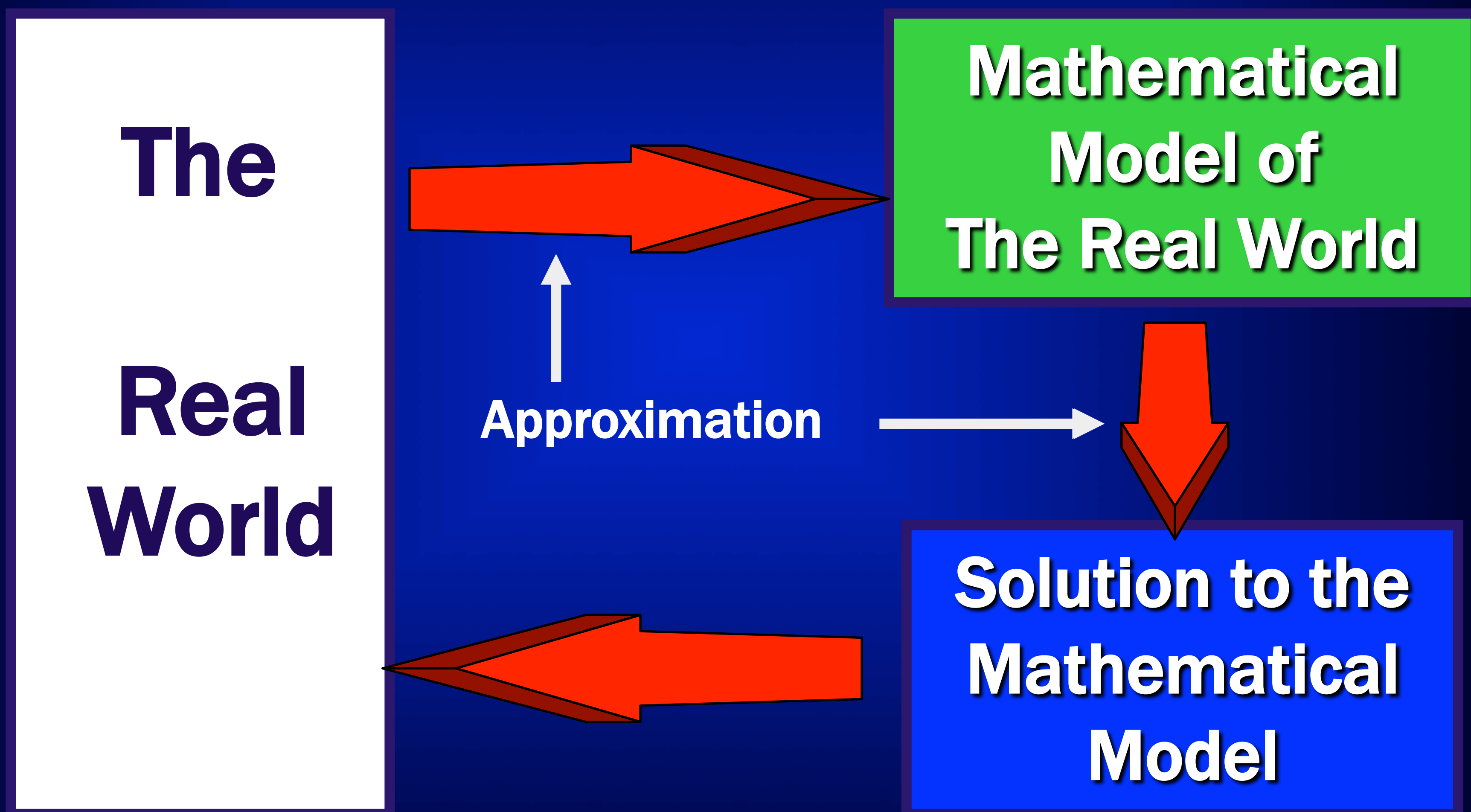
- Hamming once asked me,

"What progress of today will be remembered 1000 years from now ?"

Let's simplify it: ***Will your work be remembered 100 years from today?***

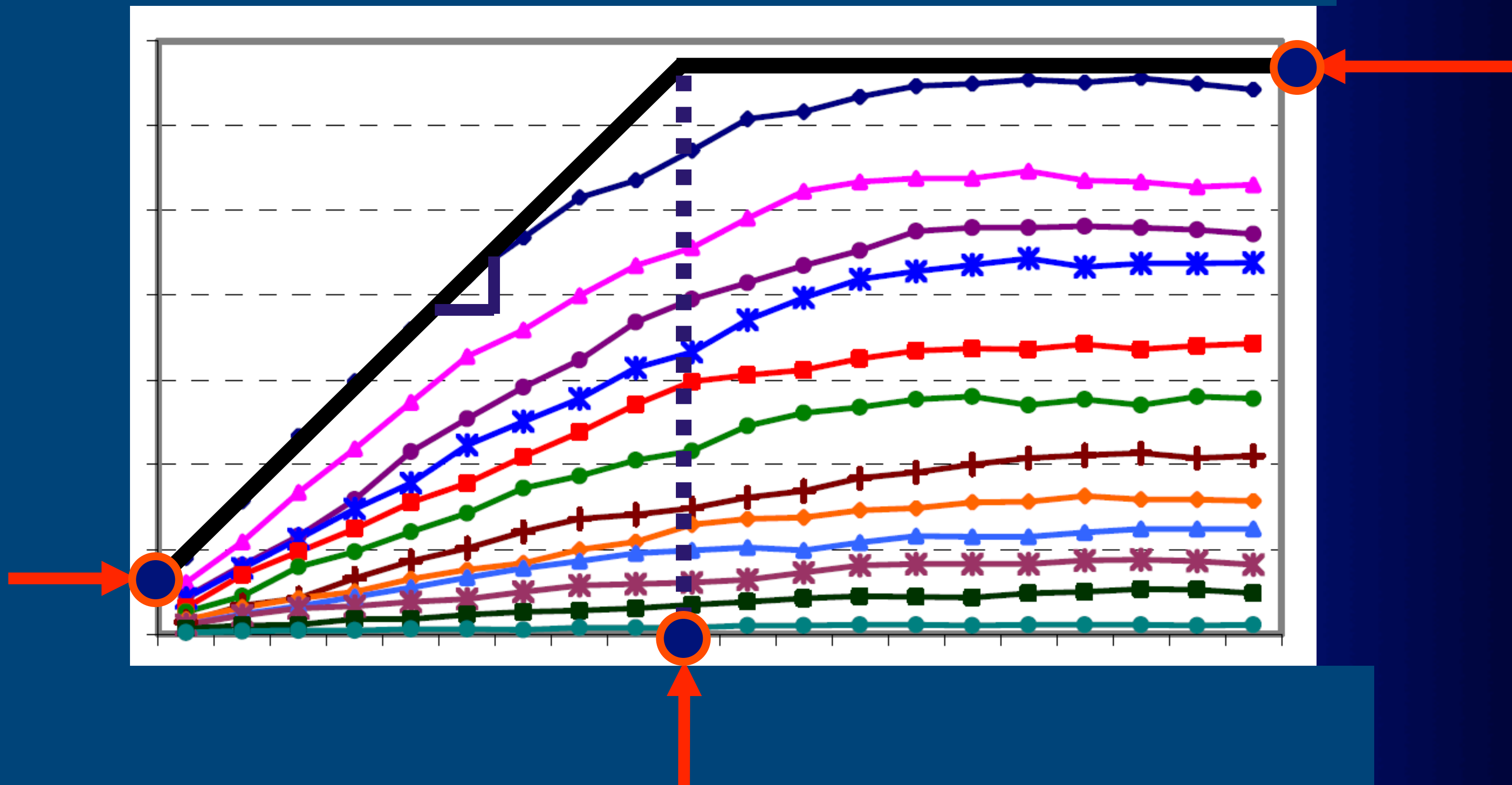
©Leonard Kleinrock 2014

2. But Don't Fall in Love With Your Model



©Leonard Kleinrock 2014

3. Beware of Mindless Simulation
Ask the Obvious Questions



©Leonard Kleinrock 2014

4. Understand Your Own Results

- Take the time to think deeply about your results.
- Use deterministic or simple models to explain behavior
  - e.g. why does "filling the pipe" make sense
- Think about upper and lower bounds
- Take limits to force behavior
- Look at extreme cases to check validity and intuition

©Leonard Kleinrock 2014

5. Look for "Gee, that's funny!"

- Don't ignore strange looking results
  - Often that's where the "gold" lies
- The greatest scientific discoveries are Not accompanied by "Eureka", but most occur when someone mutters, "*That's* interesting"

©Leonard Kleinrock 2014

More on Modeling

- Moving the frontier is tough (we mislead our students)
- Once they move it, they will be able to repeat it again (students don't believe us)
- Teach your students to understand their results!
- Generalization usually comes when you can see the simplicity of a solution
- As Norbert Wiener said, "Every scientist must occasionally turn around and ask not merely "How can I solve this problem?" but, "Now that I have come to a result, what (other) problems have I solved?"
- When a field gets too crowded, move your research vector slightly
- Keep your interest in related areas, areas where something might happen.

©Leonard Kleinrock 2014

Thank You

www.lk.cs.ucla.edu

©Leonard Kleinrock 2014