Energy & QoS in Systems & Networks

Erol Gelenbe

www.ee.imperial.ac.uk/gelenbe

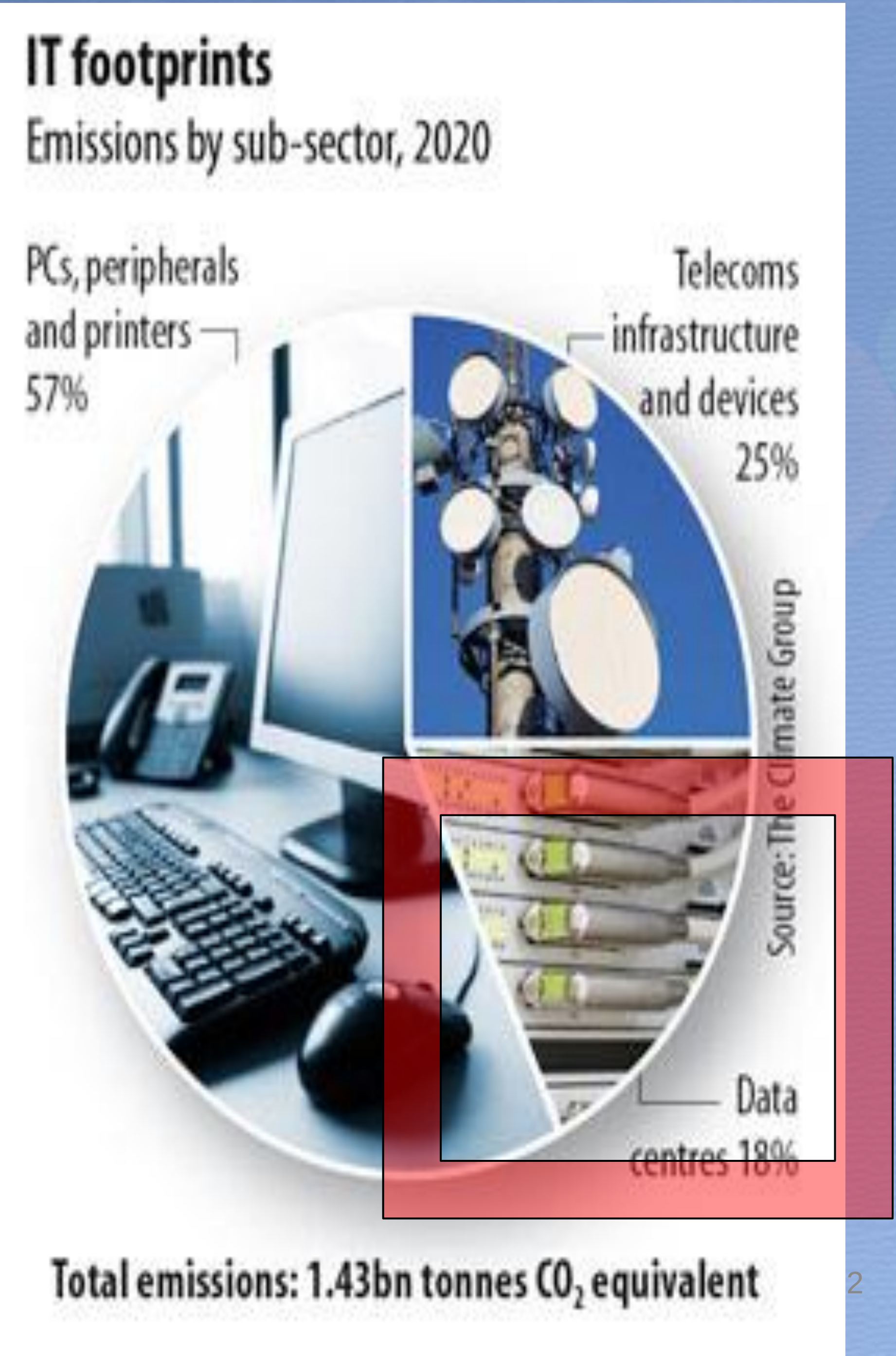
Intelligent Systems & Networks Group
Electrical & Electronic Eng'g Dept
Imperial College
London SW7 2BT

2020 ICT Carbon → 1.43BTONNES CO_2

2007 ICT = 0.83BTONNES CO_2

~ Aviation = 2%

Growth 4%



360m tons CO_2

260m tons CO_2

Imperial College London

EU 2012 → ICT = 4.7% of Electricity Worldwide

D8.1: Overview of ICT energy consumption

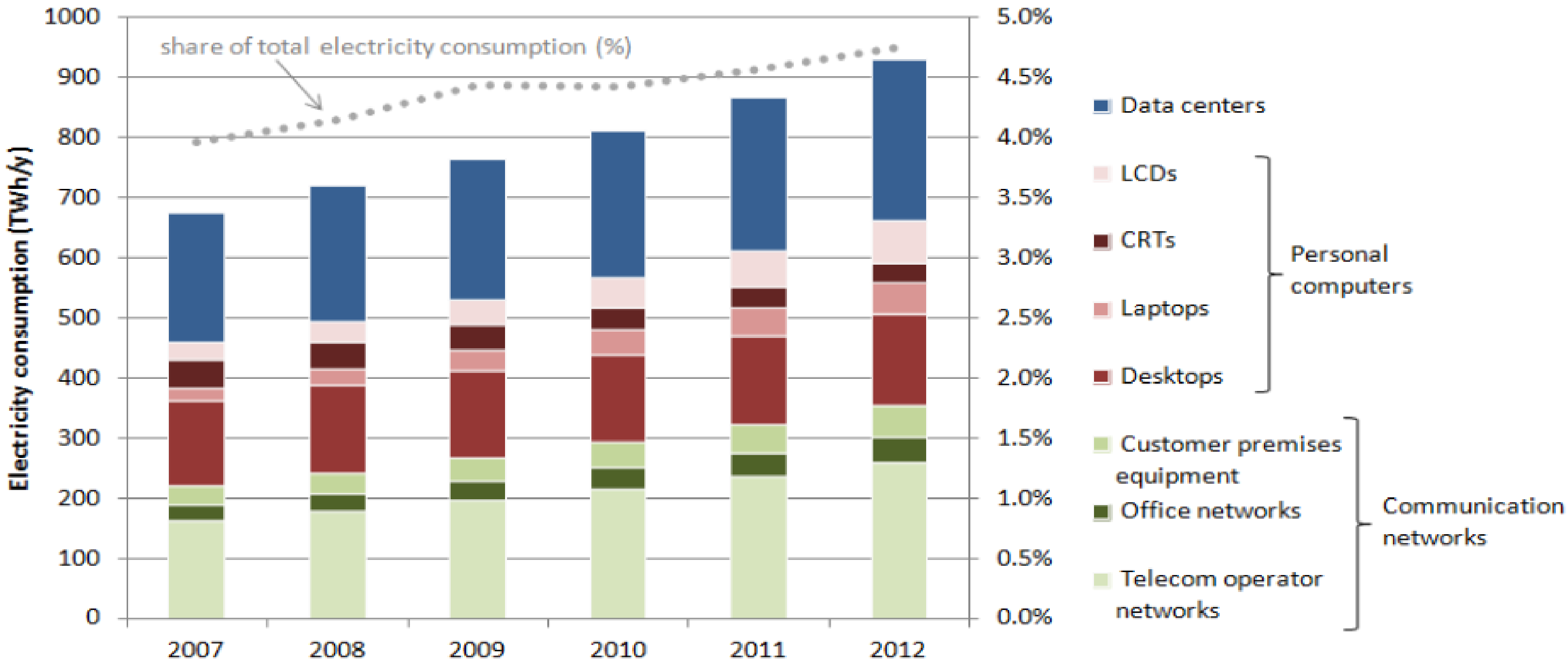
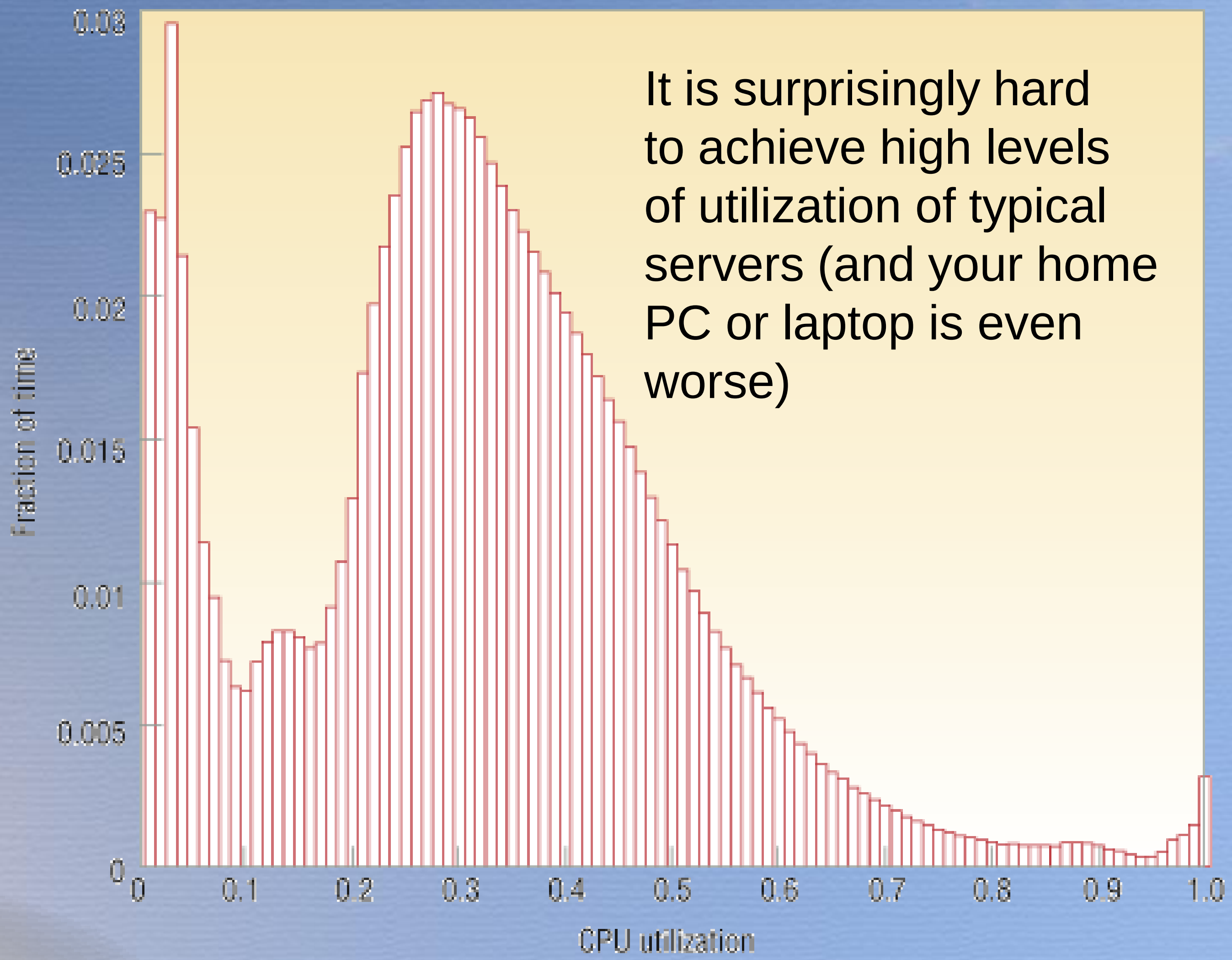


Figure 3-1: Worldwide use phase electricity consumption of communication networks, personal computers and data centers. Their combined share in the total worldwide electricity consumption has grown from about 4% in 2007 to 4.7% in 2012.

Imperial College London

Computing Loads are Generally Low

"The Case for Energy-Proportional Computing,"
Luiz André Barroso,
Urs Hölzle,
IEEE Computer
December 2007

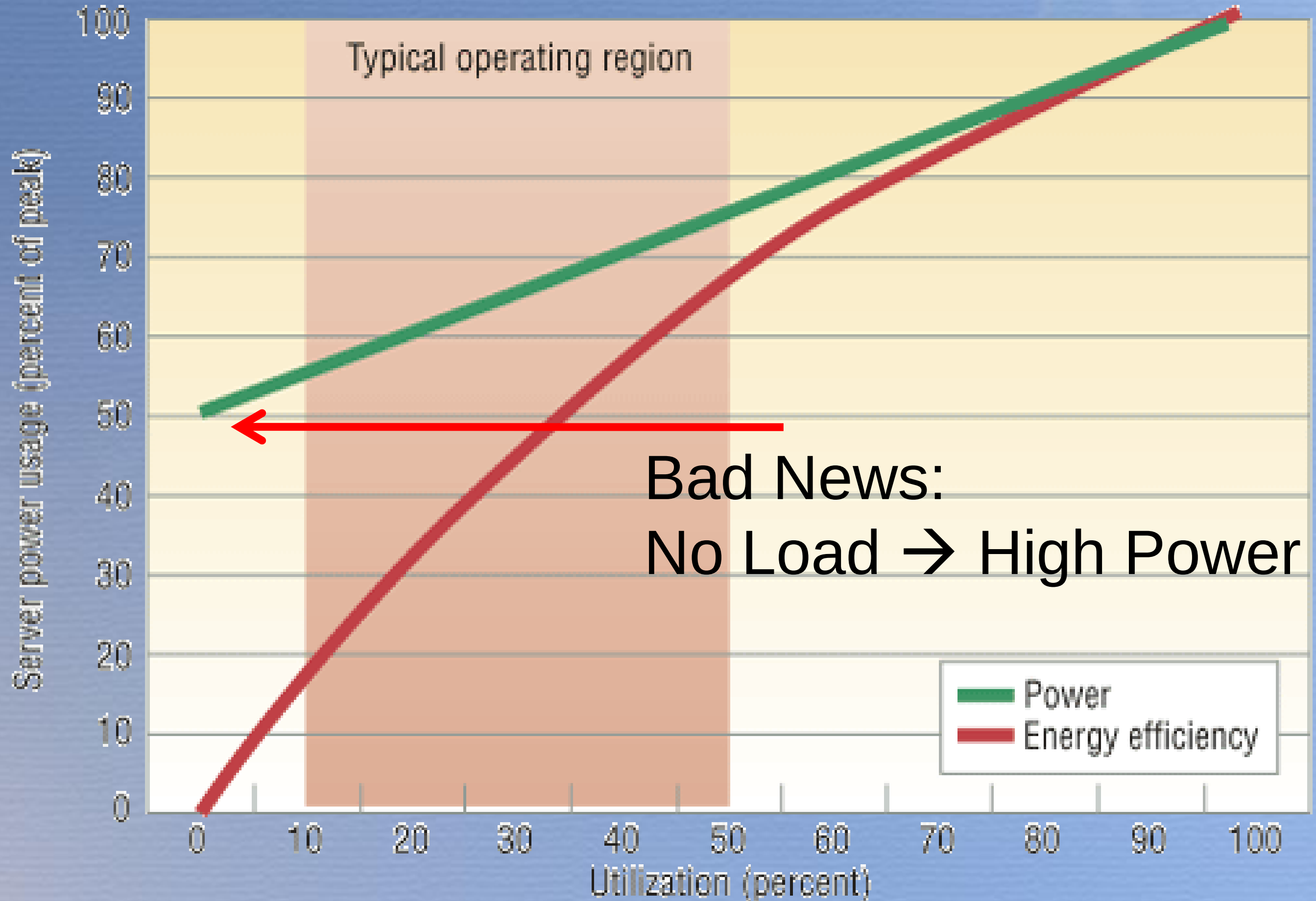


It is surprisingly hard to achieve high levels of utilization of typical servers (and your home PC or laptop is even worse)

Imperial College London

Energy Consumption at Low Loads Remains High

"The Case for Energy-Proportional Computing," Luiz André Barroso, Urs Hölzle, *IEEE Computer* December 2007

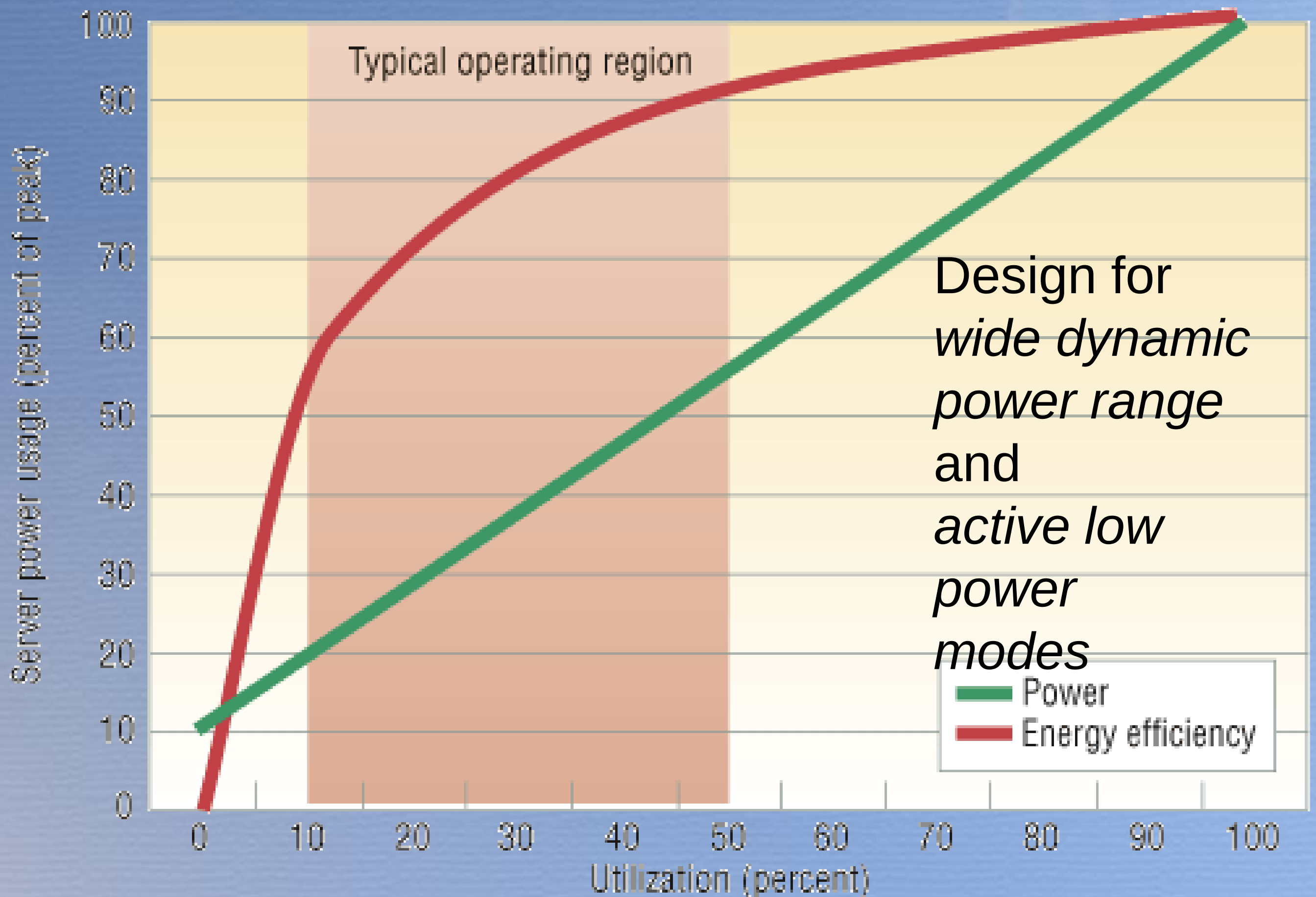


Energy Efficiency = Machine Utilization/Power

Imperial College London

Energy Proportional Computing

"The Case for Energy-Proportional Computing,"
Luiz André Barroso,
Urs Hölzle,
IEEE Computer
December 2007



Design for *wide dynamic power range* and *active low power modes*

Energy Efficiency = Server Utilization/Power

Imperial College London

Is this Socially Acceptable & Sustainable?

Estimated Added Value of ICT

5~7% = CO2 Savings/ ICT CO2 Emissions

- Google … and Other Myths
  - 0.3Wh per « Google search »
  - Facebook: 500Wh / User / Year
  - Energy Costs $ can be as High as 15% of ICT Operational Costs
  - US Costs are 40% Less than UK and 70% Less than Germany (Canada ?)

Mobile and Intermittent Computing ??

Load Averaging in Space and Time??

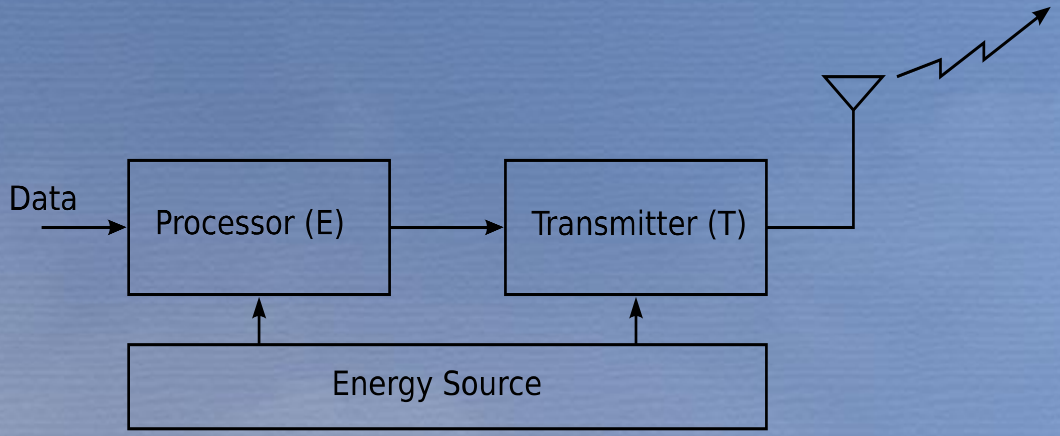
Re-Use Heat ?? New Wireless Business Model??

Imperial College London

Our Work on Energy and ICT

- **Energy Aware Ad Hoc Networks (2004)**
- **Wired Network test-bed to seek way forward (2009)**
- **Wired Energy-Aware Software Defined Network (2010-11)**
- **QoS-Energy Aware routing algorithms (2010-12)**
- **Energy and Time Trade-Offs in Internet Search (2010-13)**
- **Energy-QoS Trade-Offs in Servers and Clouds (2010-13)**
- **Micro & Nano-Scale (2013-2016)**
- **EU Projects: EU FP7 Fit4Green, ERA-NET ECROPS …**

Wireless: EPSRC ECROPS Project (2013-2016)

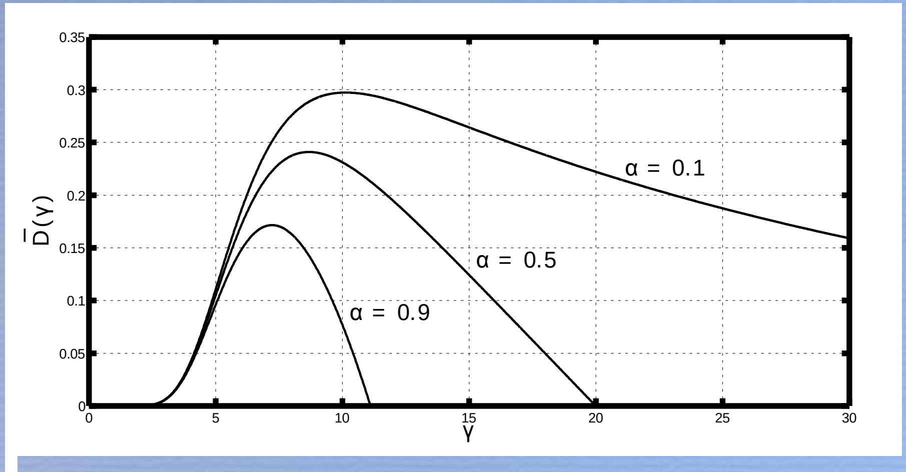


Effective Transmission Time
VS Power P_T

$$D_{eff} = \frac{D}{f\left(\frac{r P_T}{B + I}\right)}.$$

Number of Bits Correctly Transmitted per Units of Power

$$\bar{D}(P_T) = \frac{f\left(\frac{r P_T}{B + \alpha P_T}\right)}{P_E + P_T}.$$



Imperial College London

Energy Efficiency and Computer Systems

- **Ideal: Power Proportional to Utilisation**

$$\Pi = \omega\rho$$

$$\text{energy consumption per job in joules}$$

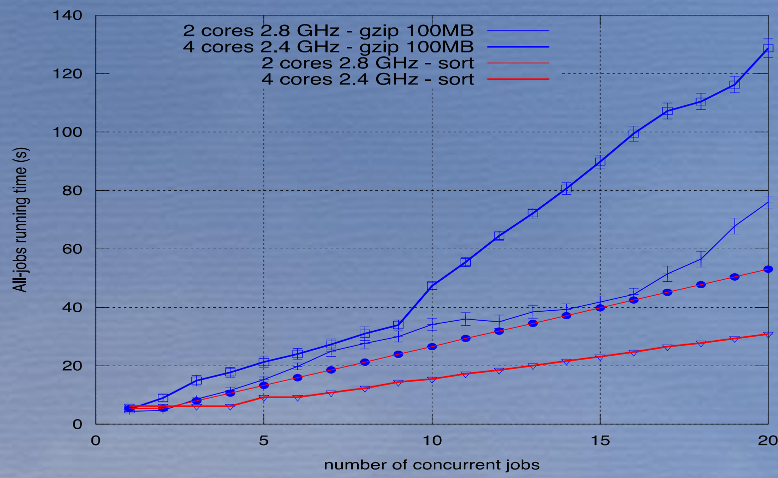
$$J_{job} = \Pi/\lambda = \omega E[S]$$

- **Reality is Different**

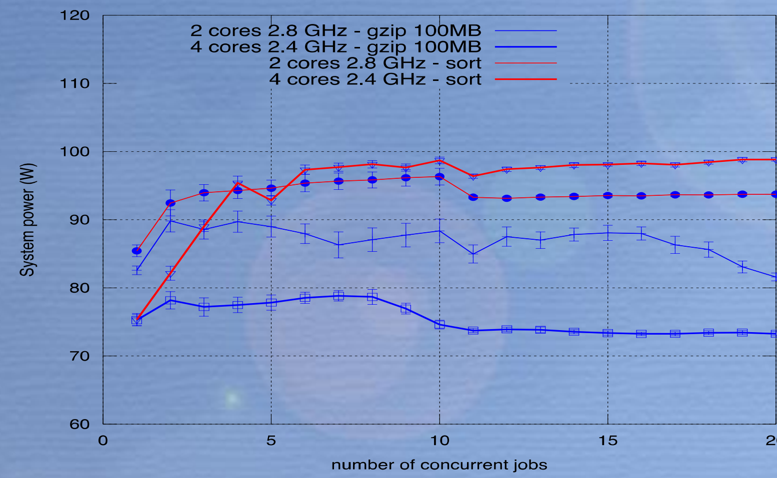
$$\Pi = A + B\rho \qquad J_{job} = \frac{A}{\lambda} + BE[S]$$

Imperial College London

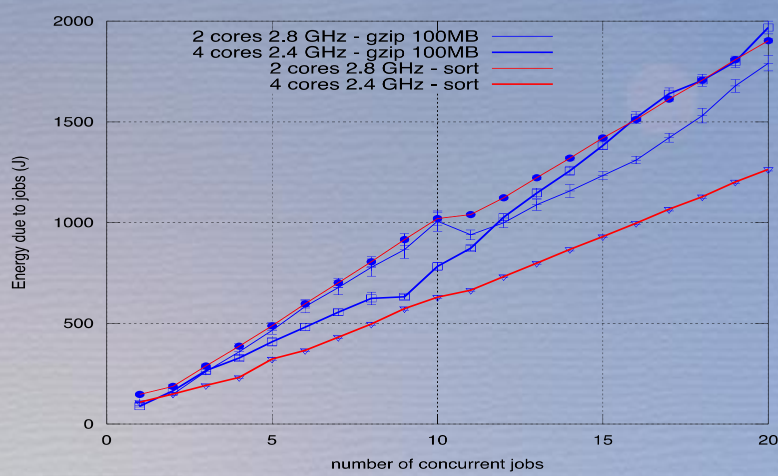
Power for Compute-Intensive Apps



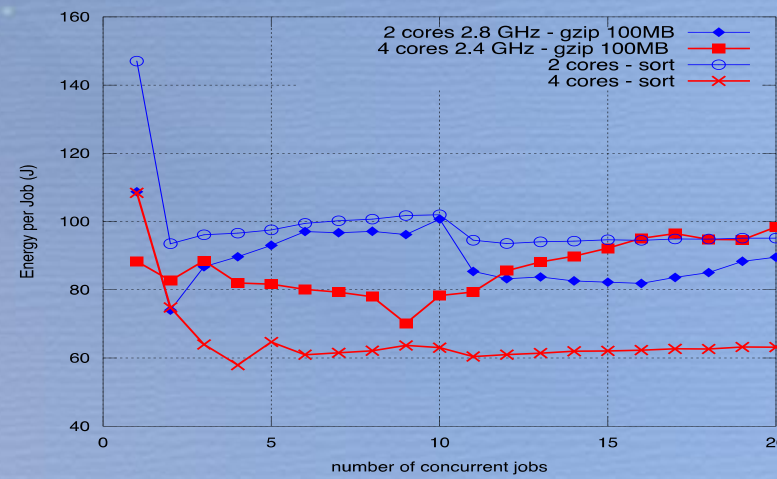
(a) Running time.



(b) Measured power consumption.



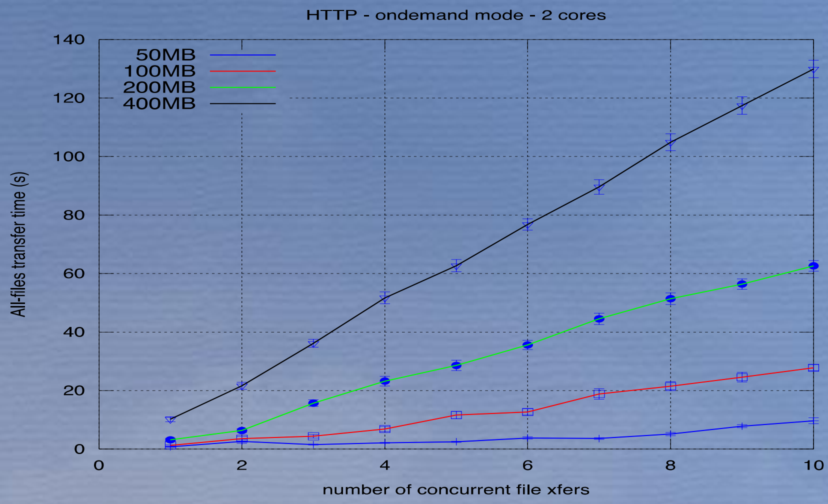
(c) Measured energy consumption.



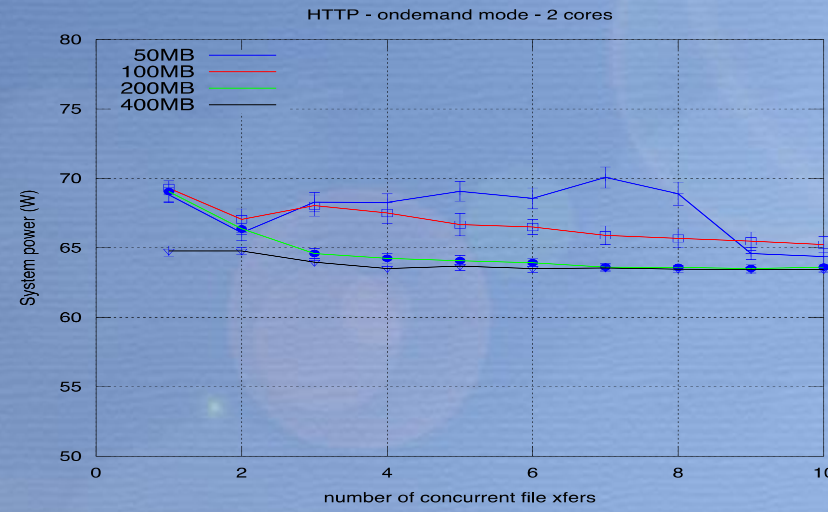
(d) Energy per job.

Imperial College London

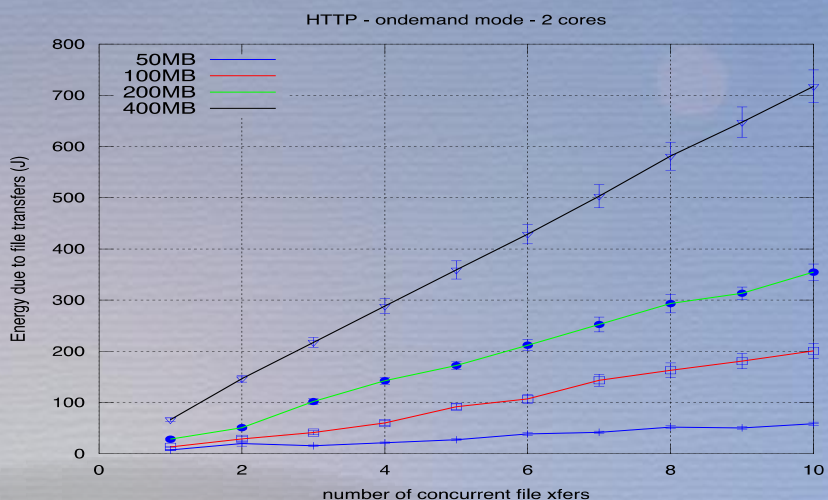
Power in Network Intensive HTTP



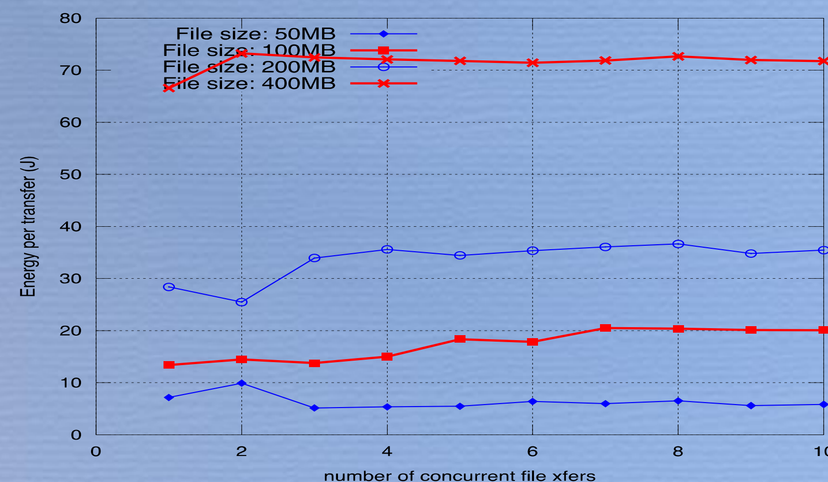
(a) Running time.



(b) Measured power consumption.



(c) Measured energy consumption.



(d) Energy per job.

Imperial College London

Simple Composite Cost C_{Job} for Delay and Energy

- **Composite Cost Function:**

a.[Average Response Time per Job]

+ b.[Average Energy Consumption per Job]

$$
\begin{aligned}
C_{job} &= \frac{aE[S]}{1-\lambda E[S]} + bJ_{job} \\
&= \frac{aE[S]}{1-\lambda E[S]} + \frac{bA}{\lambda} + bBE[S]
\end{aligned}
$$

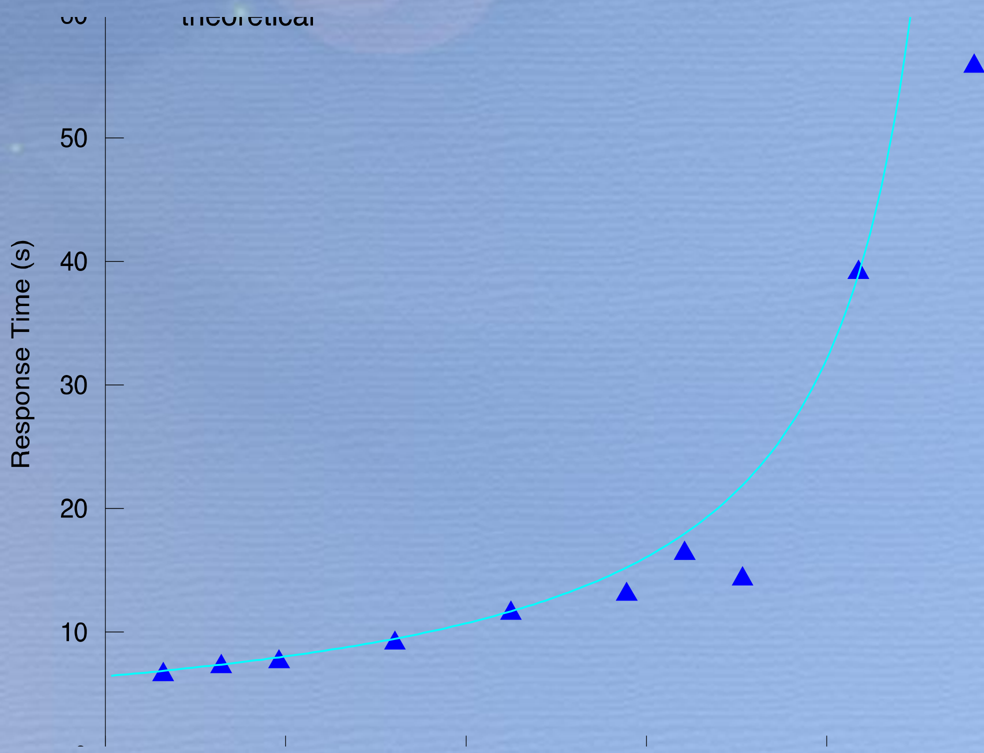
Imperial College London

Measurements

To validate the energy-QoS metric and optimum load model, we conducted a series of experiments using jobs executing on a server class system having a quad-core Intel Xeon 3430 (8M cache, 2.4 GHz), 2 GB RAM, single 150 GB SATA hard drive, and 2 on-board Gigabit Ethernet interfaces. The system runs Linux (Ubuntu) with CPU throttling enabled with the *ondemand governor*, which dynamically adjust the cores' frequency depending on load. A client machine is attached to the server through a fast Ethernet switch to generate the workload, and the client machine also measures the system's power consumption [].

We measured power consumption when it is idle, i.e. when it has no external jobs to execute, to be $A = 69.5$ Watts, which corresponds to the value of A in equation (4).

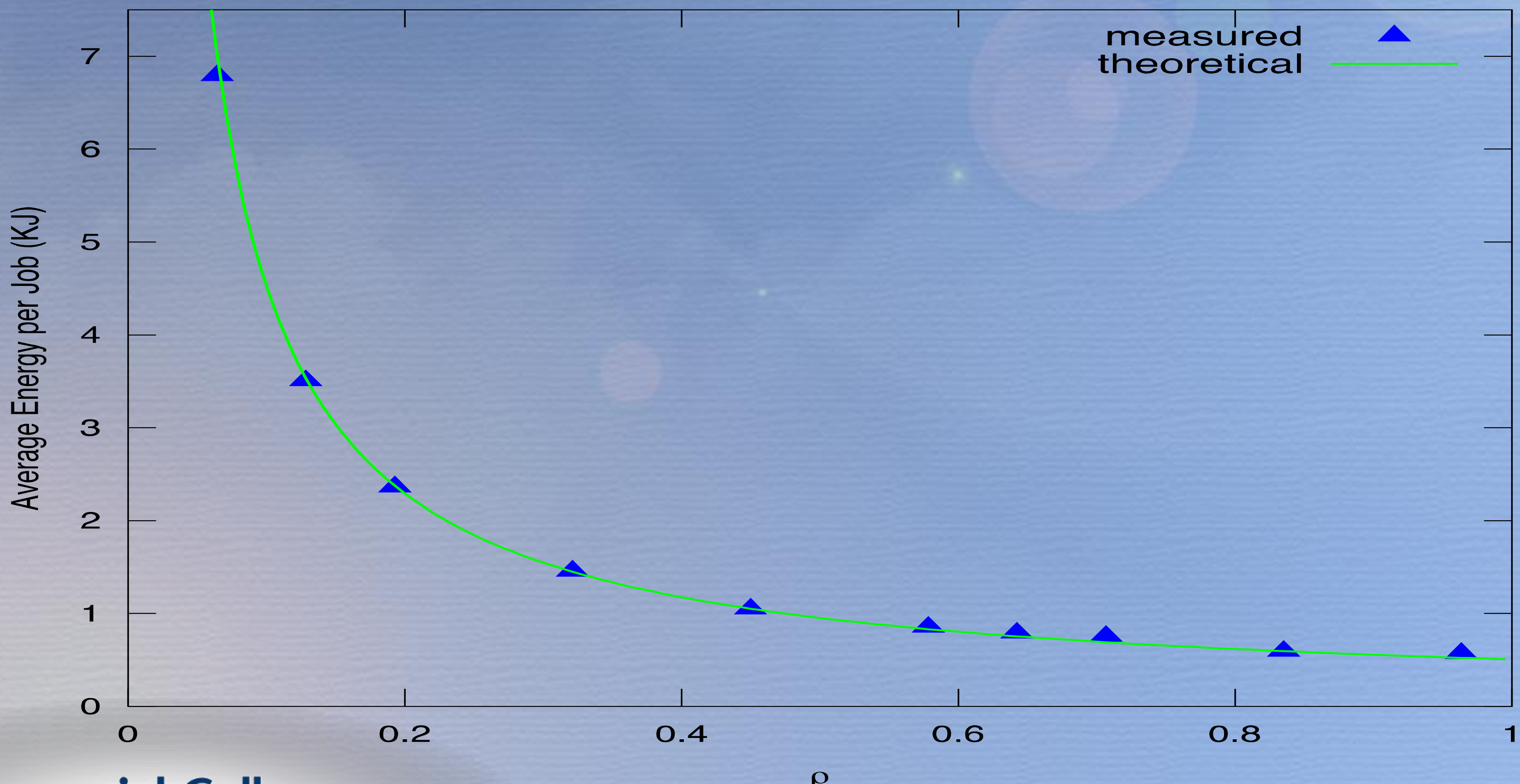
Then we measured the average energy consumed by a single job from observations obtained from serving a large number of jobs (1000), the average power consumption and the total running time of the experiment. The value of B was measured to be 13.24 Watts per job on average. The measured value of J_{job} and the calculated results from (4) we the experimentally estimated values of A and B are shown



Imperial College London

Validation

- **Average Energy Consumption per Job vs Load**



Imperial College
London

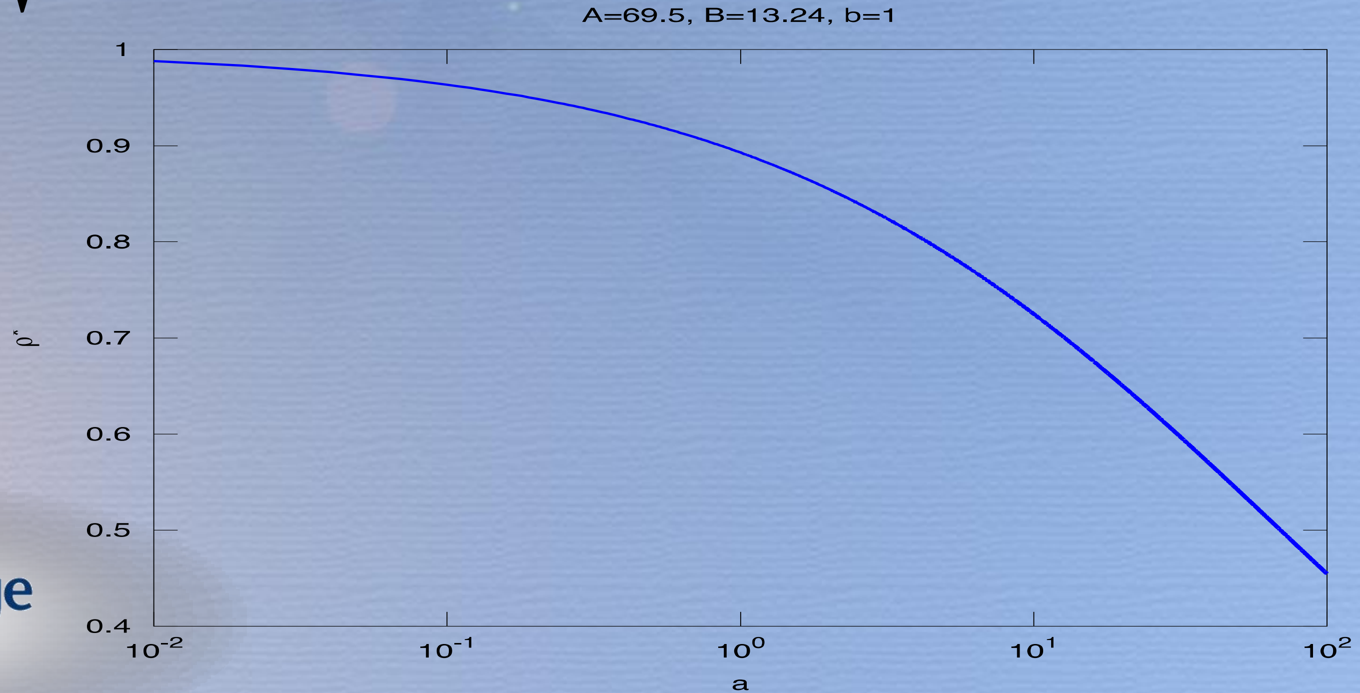
Optimisation of the Load

Optimum Load that Minimises the Composite Cost

$$\rho^* = \frac{\sqrt{\frac{bA}{a}}}{1 + \sqrt{\frac{bA}{a}}}$$

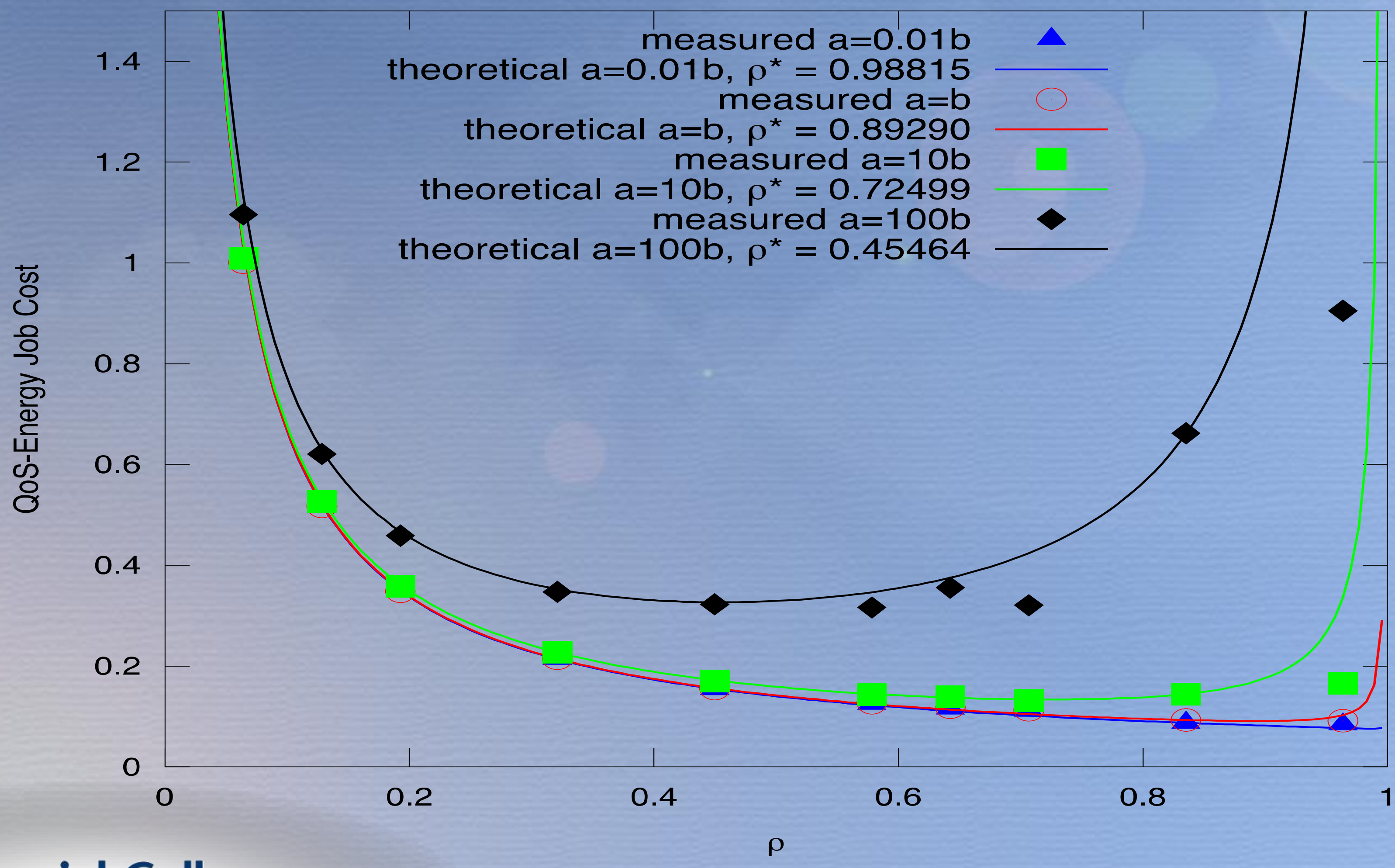
We can also see that ρ^* is an increasing function of ratio bA/a. In particular id we call $x = bA/a$ we have:

$$\frac{\partial \rho^*}{\partial x} = \frac{1}{2\sqrt{x}[1+\sqrt{x}]^2}$$



Imperial College London

Theory versus Experimental Data



Imperial College London

Optimum Load Sharing among N Heterogenous Systems

- **Cost Function**

$$C_{job} = \sum_{i=1}^{N} p_i \{ \frac{aE[S_i]}{1-\lambda_i E[S_i]} + bJ_{job}^i \}$$

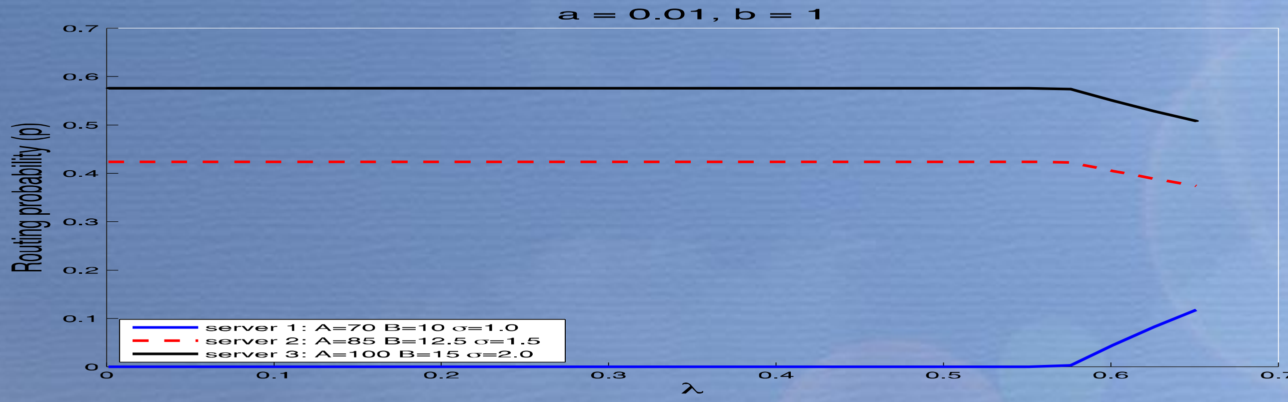
$$= \sum_{i=1}^{N} p_i \{ \frac{aE[S_i]}{1-\lambda_i E[S_i]} + \frac{bA_i}{\lambda_i} + bB_i E[S_i] \}$$

- **Optimum Load Sharing**

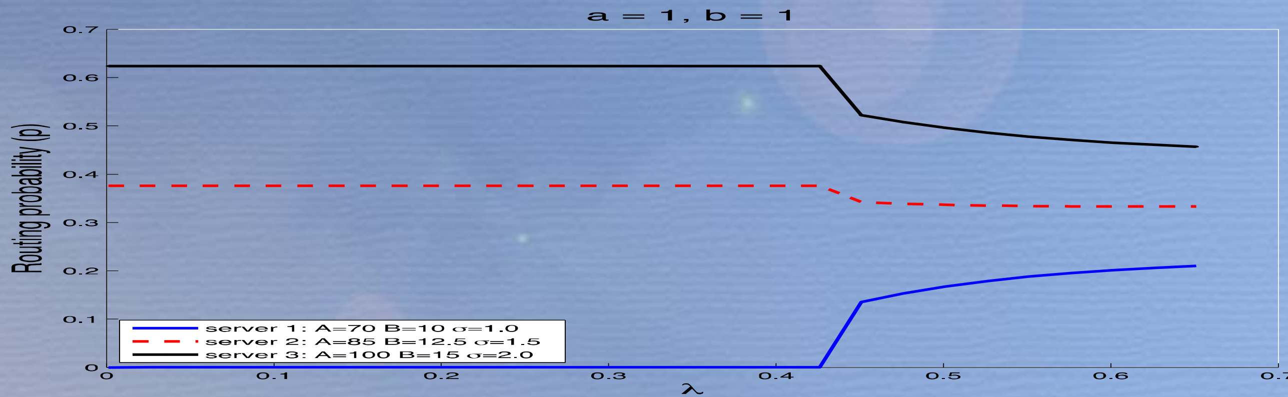
$$\rho_i = 1 - \sqrt{\frac{a}{\frac{a\sigma_i}{(1-\rho_1)^2} + b[B_1\sigma_i - B_i]}}$$

where $\sigma_i = E[S_1]/E[S_i]$ is the speed-up factor

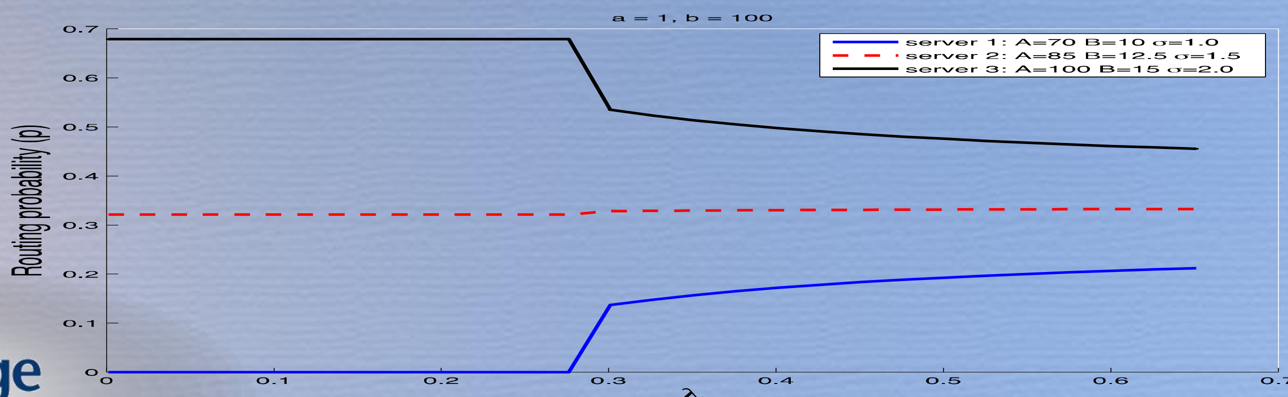
Imperial College London



(a) a=0.01



(b) a=1



(c) a=10

Imperial College London

On-Off System

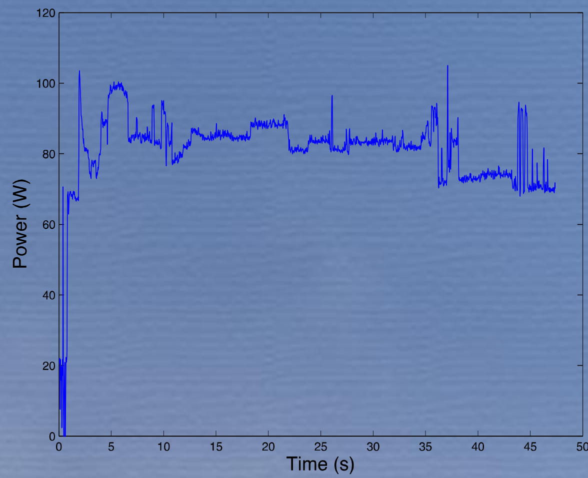
- F is the ON probability, f is the On-Off rate, γ is the On-Off Energy Consumption

$$C_{job} = \frac{aE[S]}{F - \lambda E[S]} + b\frac{FA + \gamma f}{\lambda} + bB\frac{E[S]}{F}$$

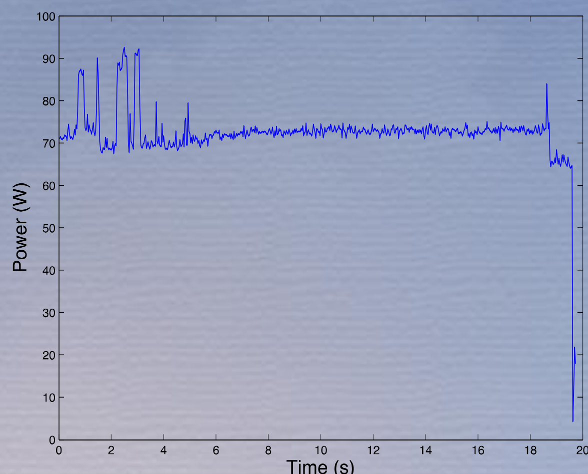
- Optimum Load is Given by

$$\rho^* = \frac{\sqrt{\frac{b(FA+\gamma f)}{a}}}{1 + \sqrt{\frac{b(FA+\gamma f)}{a}}}$$

Imperial College London



(a) System resuming



(b) System hibernating

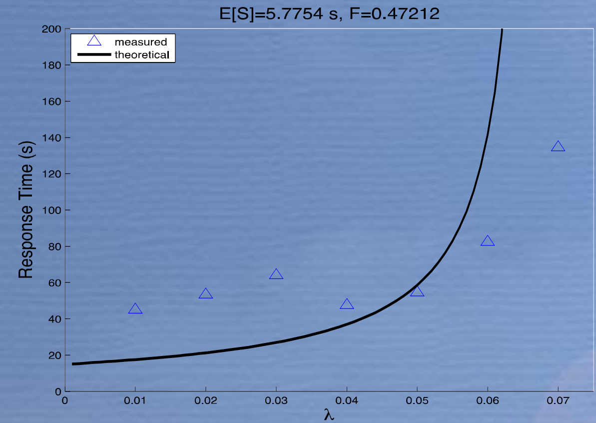


Figure 7. Measured and theoretical average response time versus load, for the system with ON-OFFs with $f = 0.005$.

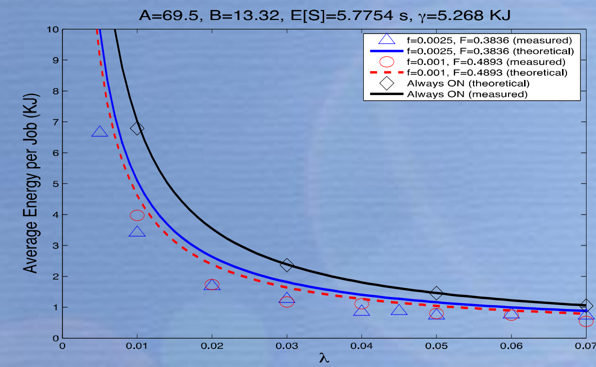


Figure 10. Theoretical and measured energy consumption per job versus load, in the system with ON-OFFs for different values of f. We se that energy can be saved when f is small and the "off" cycle is long.

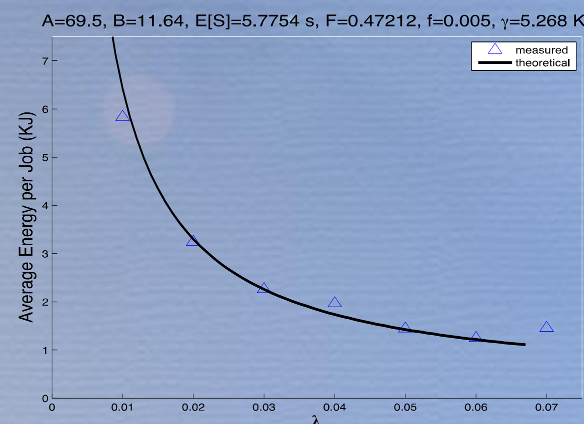


Figure 8. Theoretical and measured energy consumption per job versus load, in the system with ON-OFFs for $f = 0.005$.

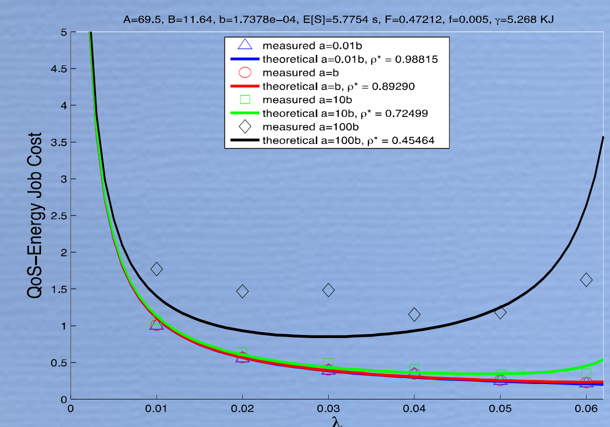
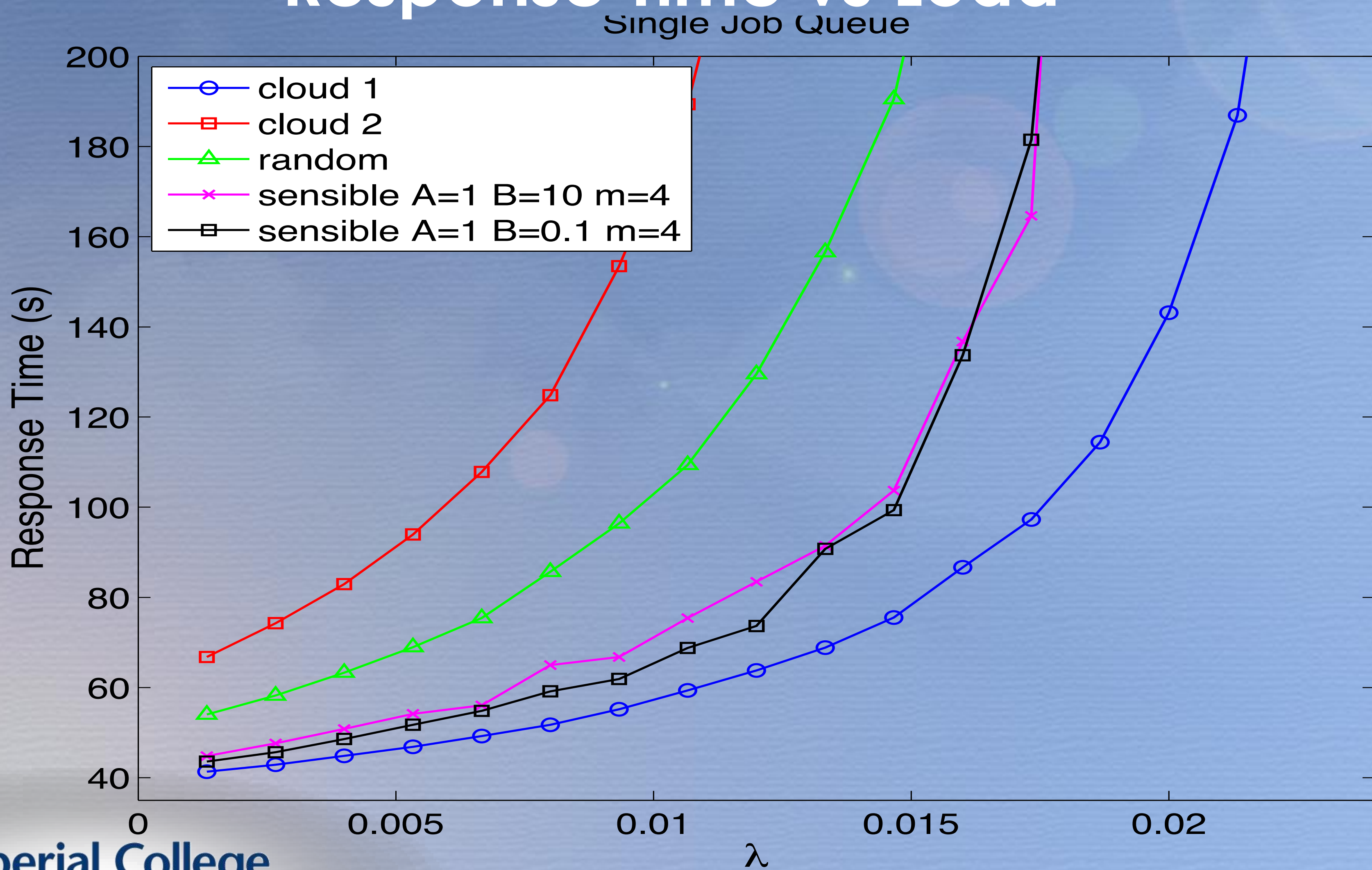


Figure 9. Composite Energy-QoS cost metric versus load in the system with ON-OFFs for $f = 0.005$.

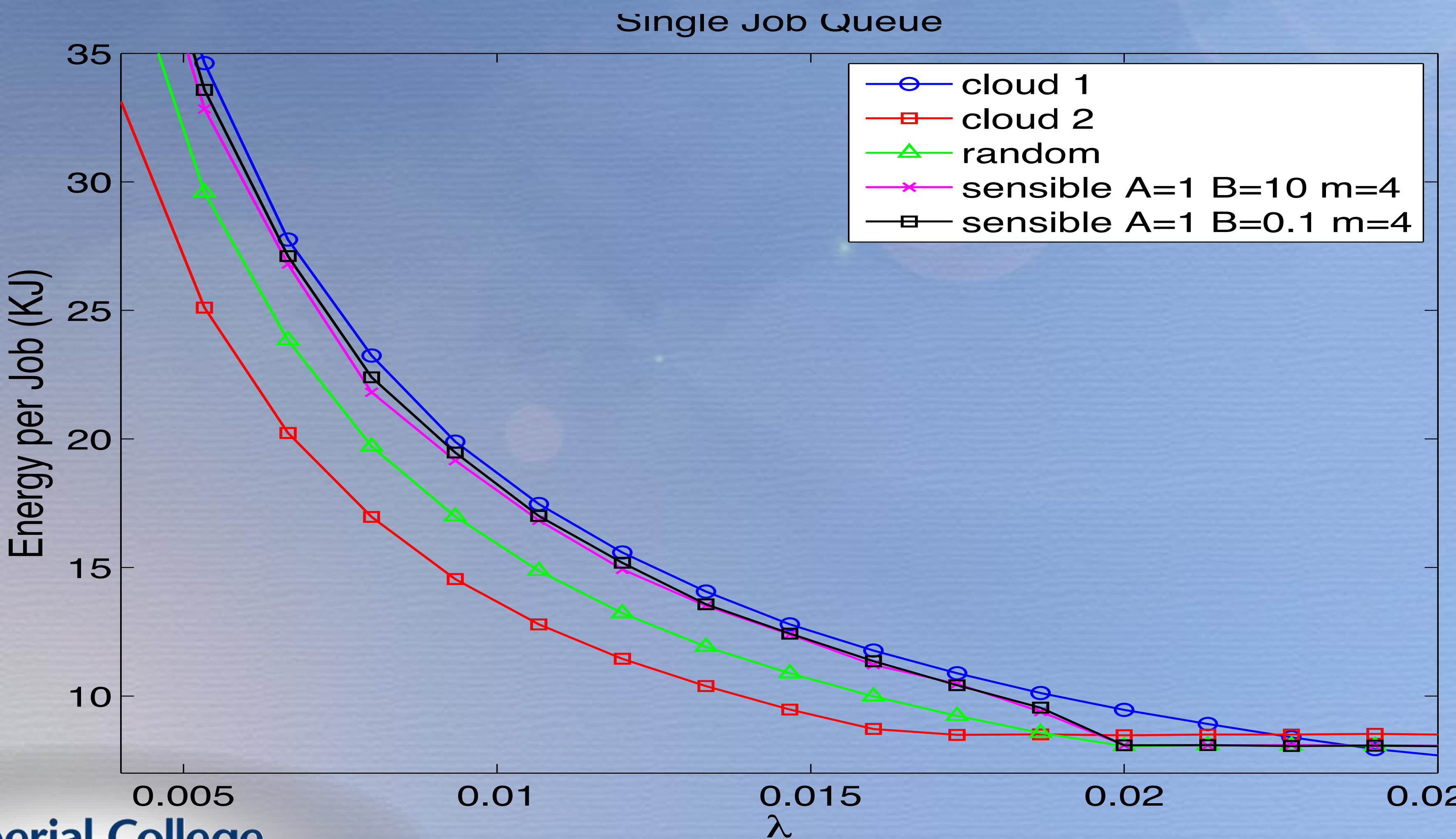
Imperial College London

Sensible Selection of a Cloud Response Time vs Load



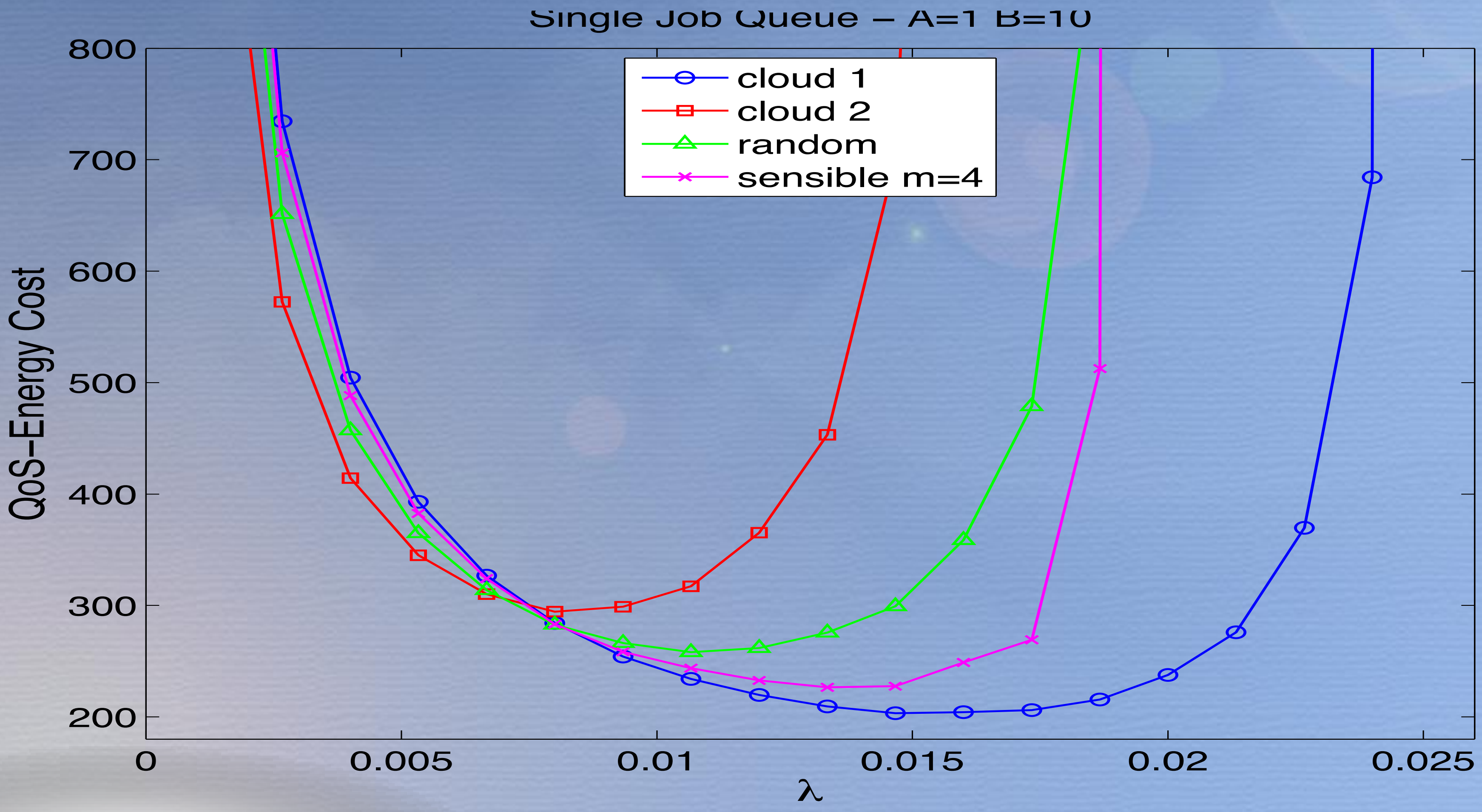
Imperial College London

Sensible Selection of a Cloud
Response Time vs Load



Imperial College London

Sensible Selection of a Cloud Composite Cost Function vs Load



Imperial College London

Energy efficiency in wired networks

- Techniques for energy savings in wireless (sensor) networks have been very widely studied
- Wired networks have been largely neglected even though they are massive consumers of power
- In a wired packet network the problem is to:
  - Minimize total power consumption, and obviously ...
  - Respect users’ QoS needs

Imperial College London

The Network Case: Experiments

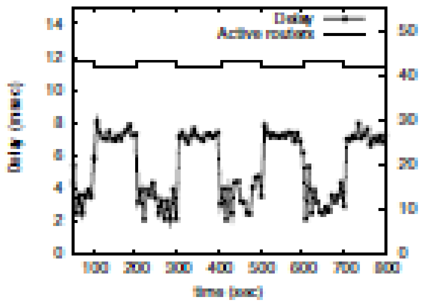
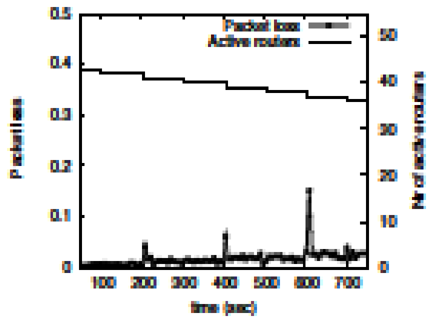


Fig. 5. Delay - second experiment



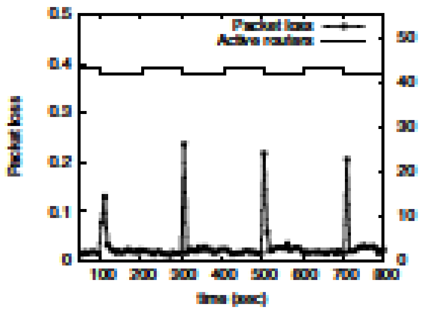
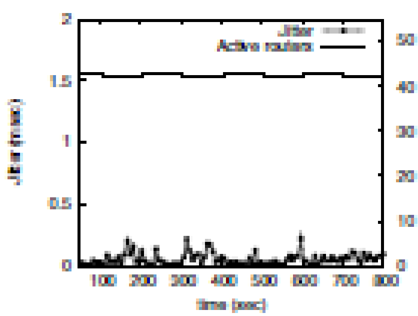
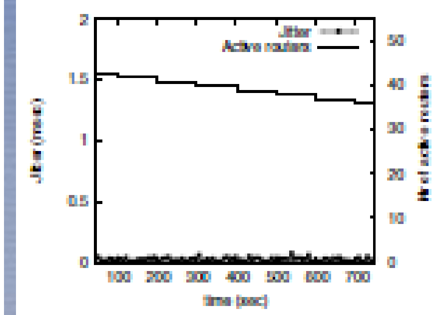


Fig. 6. Packet loss - second experiment





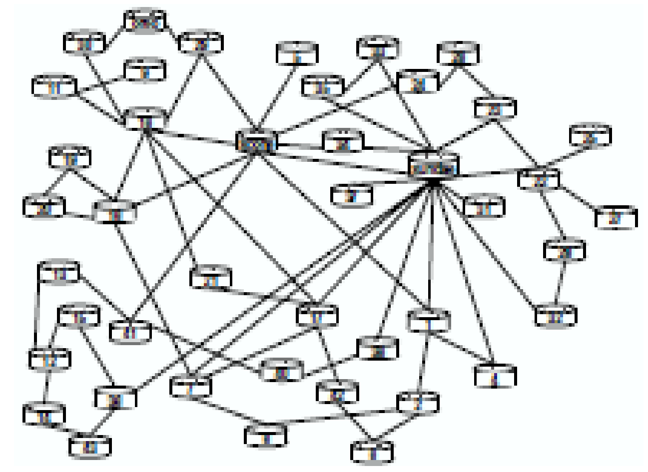
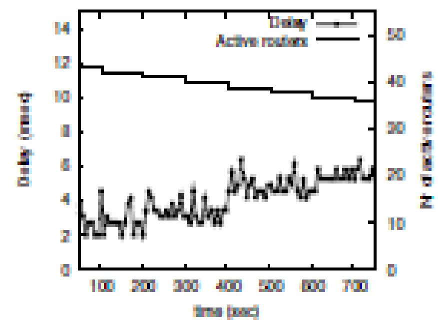


Fig. 1. Topology of the test-bed in use

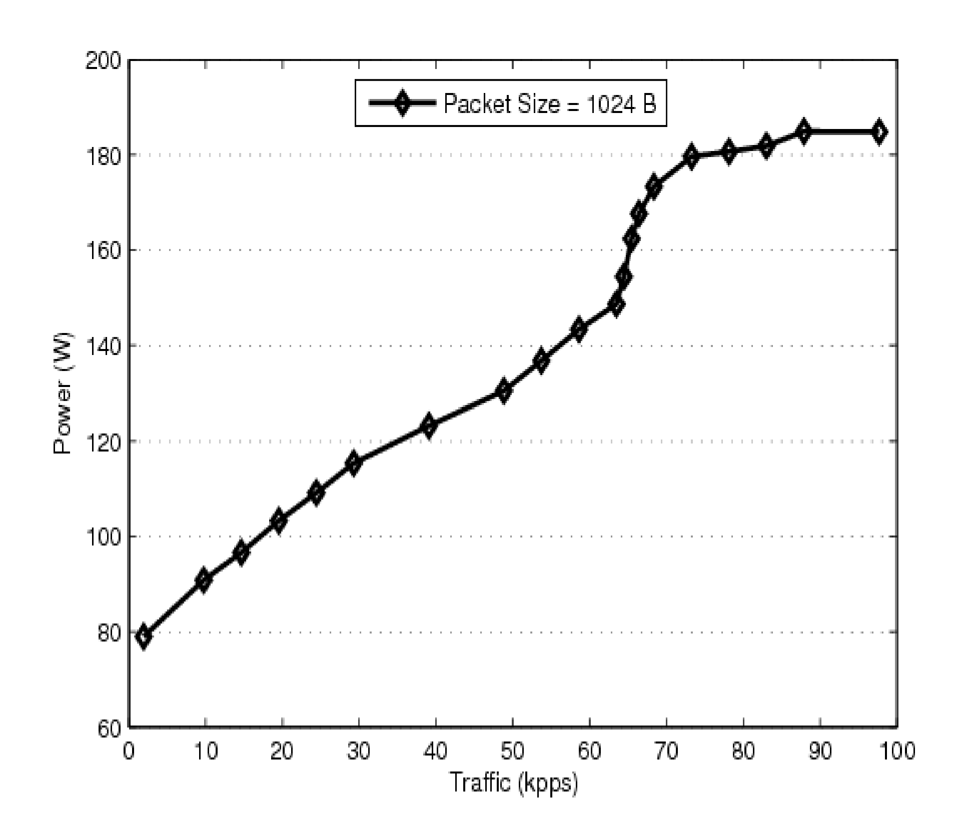


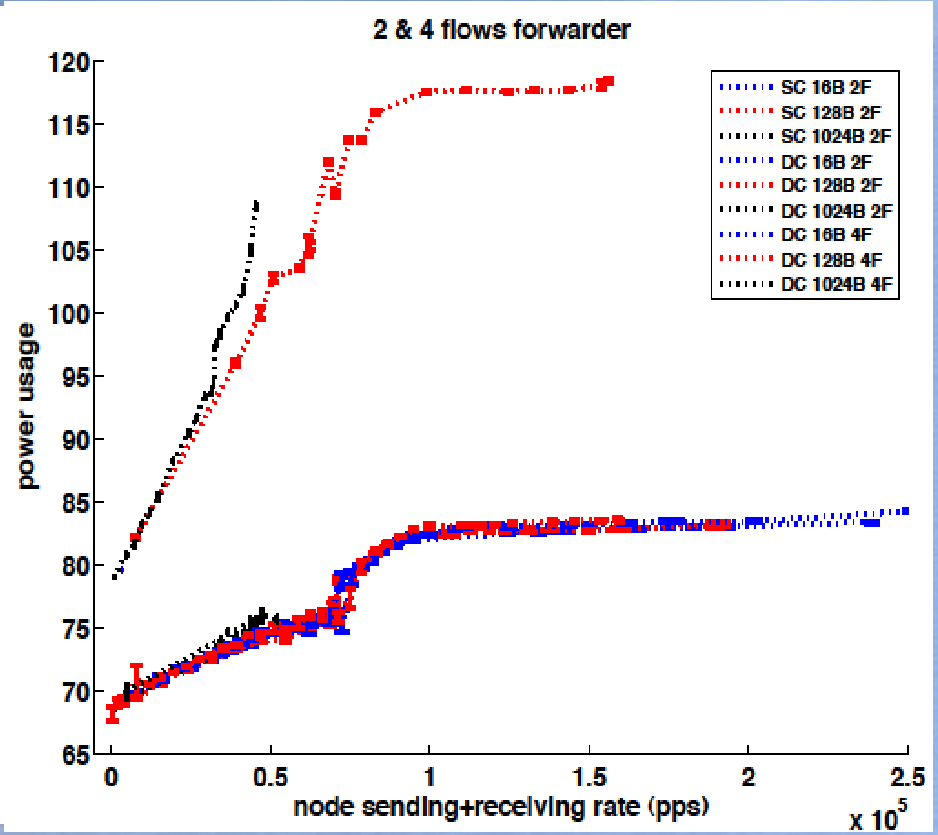
Measurements on Feasibility Using our 46-node Laboratory Packet Network Test-Bed:

E. Gelenbe and S. Silvestri, ``Optimisation of Power Consumption in Wired Packet Networks,'' Proc. QShine'09, 22 (12), 717-728, LNICST, Springer Verlag, 2009.

Imperial College London

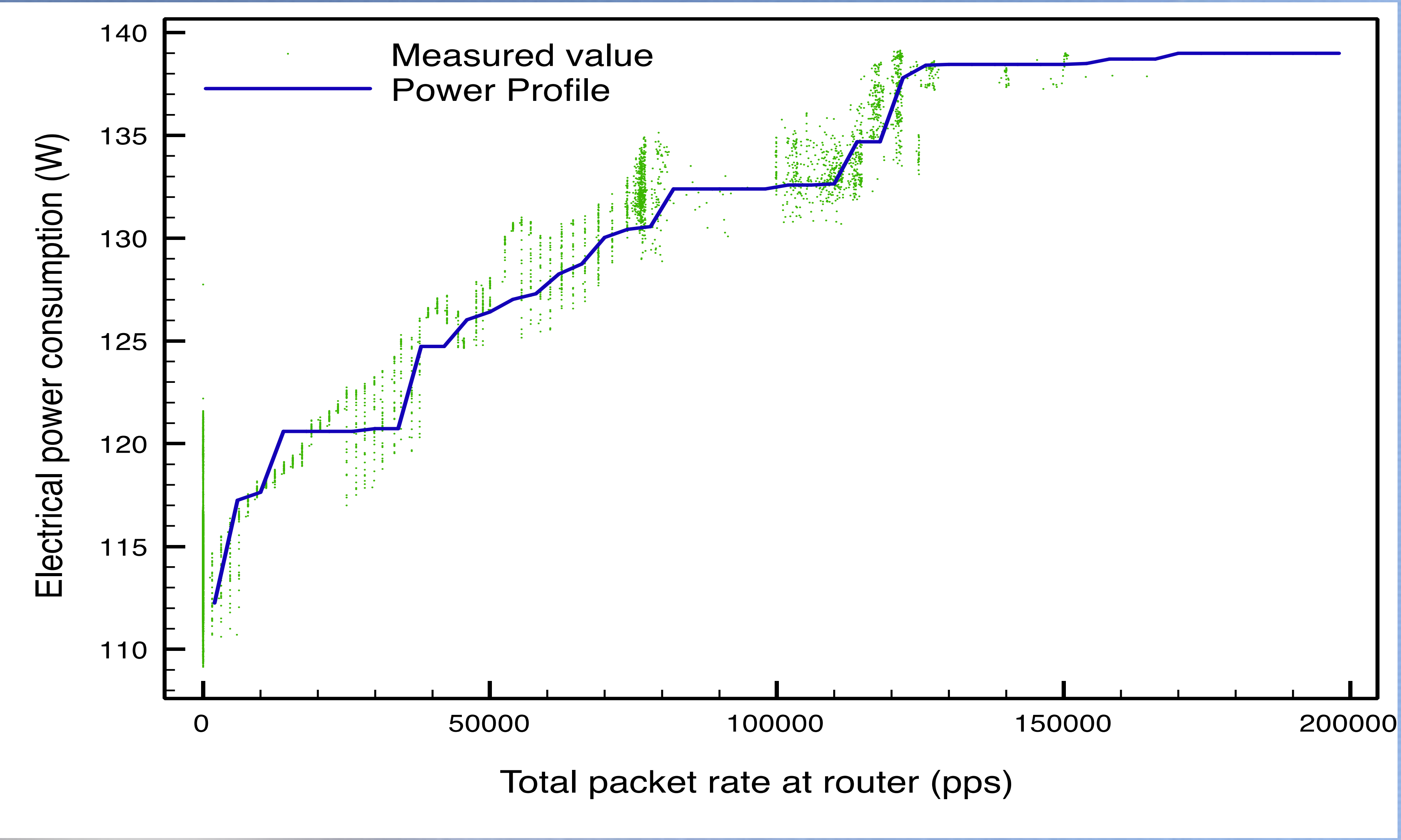
Power Measurement on Routers





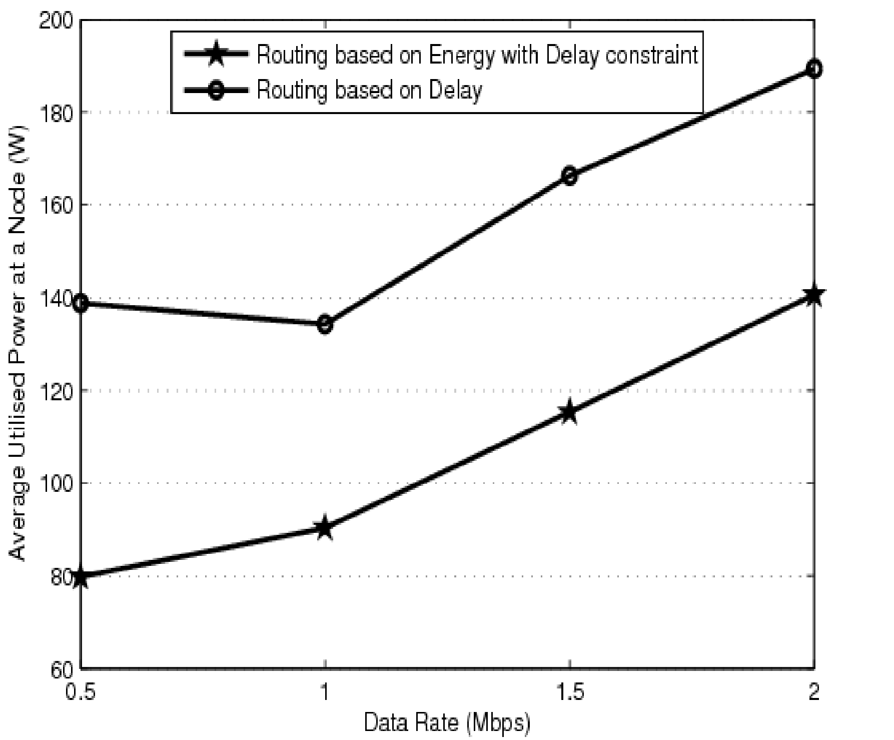
Imperial College London

Example of Measured Router Power Profile

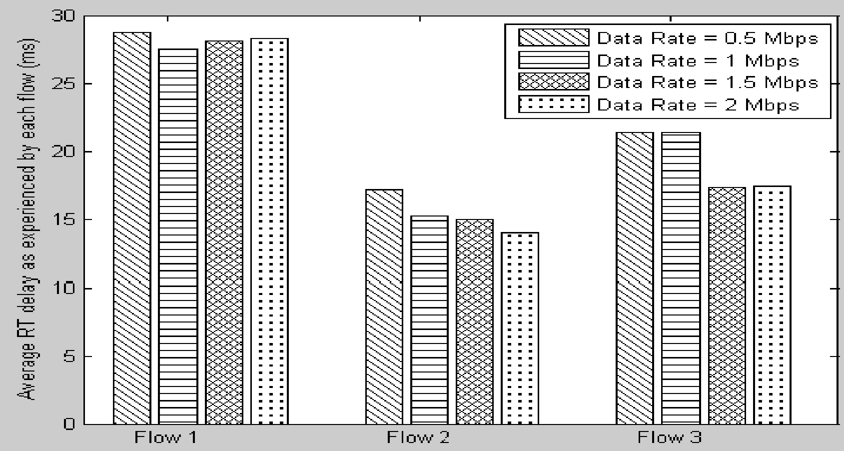


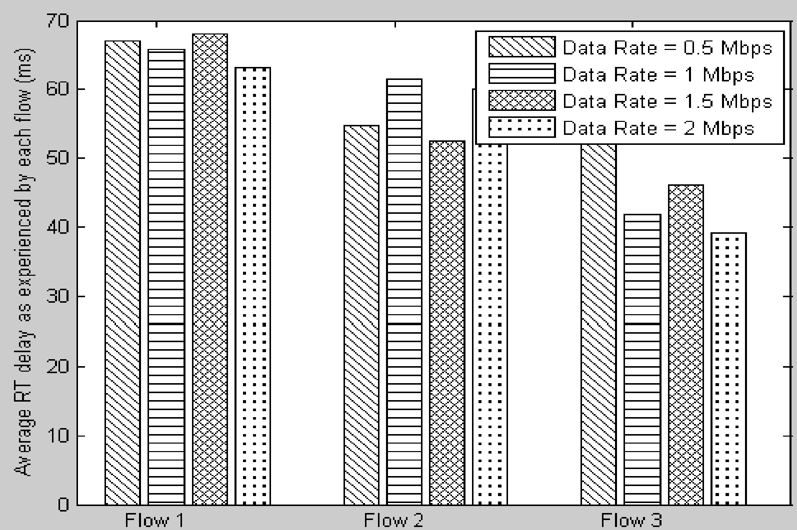
Experiments with a Self-Aware Approach
Minimise Power subject to End-to-End Delay (80ms) Constraint

[10] E. Gelenbe, ``Steps Toward Self-Aware Networks,'' Comm. ACM, 52 (7), pp. 66-75, July 2009.
[15] E. Gelenbe and T. Mahmoodi "Energy aware routing in the Cognitive Packet Network", presented at NGI/Co July 2010, submited for publication.



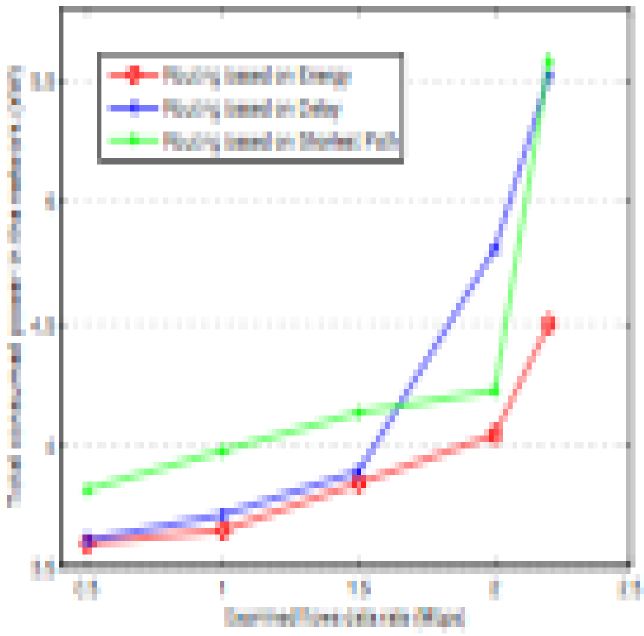
Measuring Avg Power Over All Routers Vs Average Traffic per Router



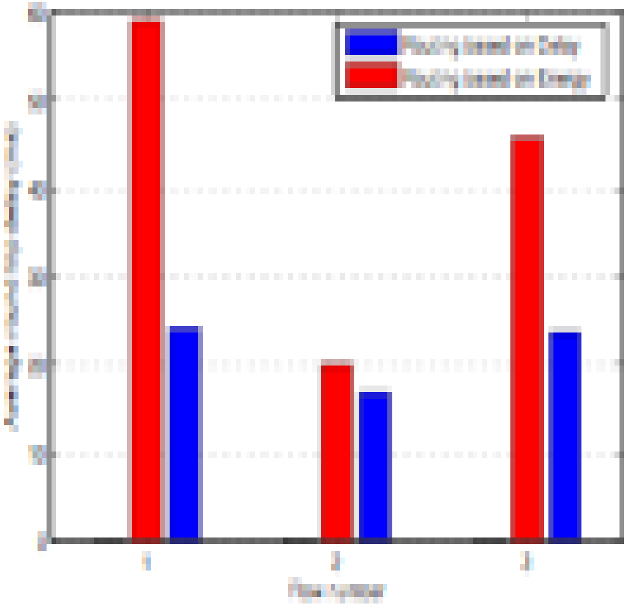


Imperial College London

Power and Delay with EARP
Energy Aware Routing Protocol



(a) Total power consumption in the network vs. traffic rate



(b) Average round trip delay

Power Savings and QoS using EARP

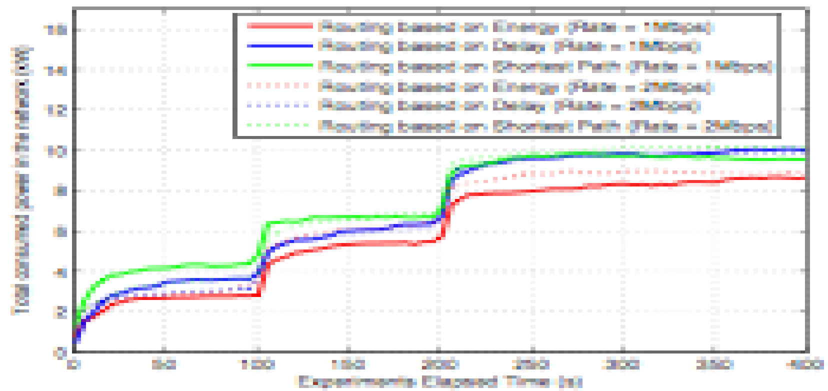
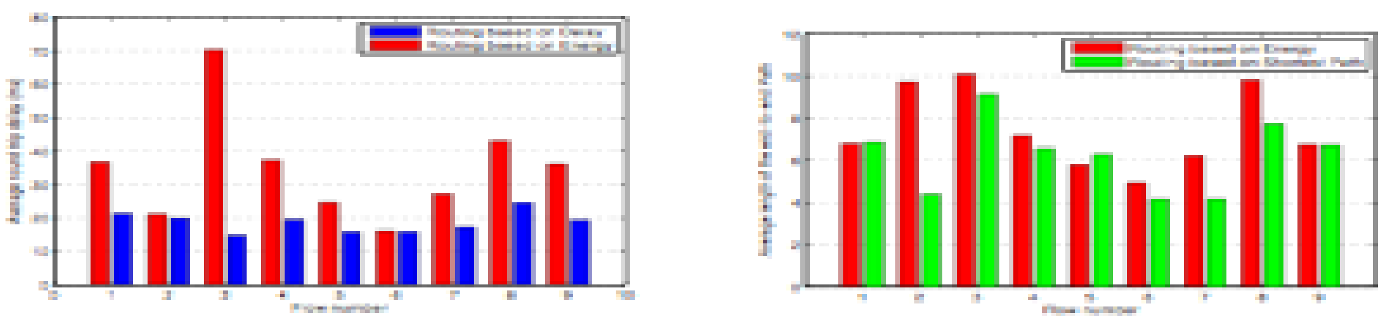


Fig. 2. Scenario two: Total power consumption in the network Vs. the experiment's elapsed time.



(a) Average round trip delay

(b) Average length of the end-to-end path

Fig. 3. Scenario two: round trip delay and the route length of the active flows.

Can Analysis and Optimisation Help for the Network Case?

IDEA:

Build a Queueing Network with Multiple Customer Classes
- A Node is a Network Router or a Network Link
- A Class is a Flow of Packets that follow the same Path

- Add Triggers to Model Control Signals that Reroute the Normal Customer Classes and also Consume Resources

Define a Cost Function that Includes Power Consumption as A Function of Load, and also A verage Response Time

- **Solve using G-Network Theory**
- **Optimise with Gradient Descent & Non-Linear Optimisation**

Imperial College London

G-networks allow product form solutions including the routing control

Rerouting controls occur infrequently (seconds) as compared to individual packet service times (1ms) and end-to-end packet travel times (10ms)

- **The system attains steady-state between the control instants**
- **G-networks [11,12,13] with triggered customer movement and multiple classes are a convenient modelling paradigm for packet networks with controls**

- **Network with N queues, R routers and L links, $N=R\cup L$**
- **Set of user traffic classes *U***
- **The default routing decision of a user of class *k* from node *i* to node *j* is represented by the probability *P(i,k,j)***
- **The external arrival rate of packets of class k to router r is denoted by *λ(r,k)***

Imperial College London

G-networks allow product form solutions that include the effect of re-routing

*C*urrent default routing decision of a user of class *k* from neighbouring queues *i* to *j* is *P(i,k,j)*

- Control traffic class *(r,k): acts at router r on traffic class k*
- A control packet of class *(r,k)* moves from queue *i* to *j* with probability *p((r,k),i,j)*
- Control function *Q(r,k,j)* : probability that user of class *k* at router *r* is directed by the corresponding control packet of type *(i,k)* to link j.
- External arrival rate of control packets of class *(r,k)* to router i : $\lambda^{-}(i(r,k))$

Imperial College London

Traffic in the Network

- **The steady state probability that a router *r* or a link *l* contains at least one packet of user class *k* is given by**

$$q(r,k)=\frac{\Lambda_R(r,k)}{\mu_r+\Lambda^-(r,(r,k))},\text{if } r\in\mathbf{R}$$

$$q(l,k)=\frac{\Lambda_L(l,k)}{\mu_l},\text{if } l\in\mathbf{L}$$

- **The total arrival rates of user packets of class k to the routers and links are given by**

$$\Lambda_R(r,k)=\lambda(r,k)+\sum_{l\in\mathbf{L}}q(l,k)P(l,k,r)\mu_l,\text{if } r\in\mathbf{R}$$

$$\Lambda_L(l,k)=\sum_{r\in\mathbf{R}}[q(r,k)P(r,k,l)\mu_r+\Lambda^-(r,(r,k))q(r,k)Q(r,k,l)],\text{if } l\in\mathbf{L}$$

Imperial College London

Control Traffic

- **The total arrival rate to router or link j of control traffic of class (i,k) is given by**

$$\Lambda^{-}(j,(i,k)) = \lambda^{-}(j,(i,k)) + \sum_{l \in \mathbf{L}} p((i,k),l,j)c(l,(i,k))\mu_l, \text{if } i, \mathrm{j} \in \mathbf{R}$$

$$\Lambda^{-}(j,(i,k)) = \sum_{r \in \mathbf{R}} p((i,k),r,j)K(r,(i,k))\mu_r, \text{if } i \in \mathbf{R}, j \in \mathbf{L}, i \neq r$$

- **The steady-state probability that a router r contains at least one packet of class k is**

$$c(l,(i,k)) = \frac{\sum_{r \in \mathbf{R}} p((i,k),r,l)K(r,(i,k))\mu_r}{\mu_l}, \text{if } l \in \mathbf{L}$$

- **And for the routers**

$$K(r,(i,k)) = \frac{\lambda^{-}(r,(i,k)) + \sum_{l \in \mathbf{L}} p((i,k),l,r)c(l,(i,k))\mu_l}{\mu_r}, \text{if } r \in \mathbf{R}, r \neq i$$

Average Queue Length

- **Each user class is assumed to be handled by separate queues in routers, so the average queue length in router *r* is**

$$N(r,k) = \frac{q(r,k)}{1-q(r,k)}, r \in \mathbf{R}$$

- **On the other hand, all packets within a link are handled in a first-come-first-serve order, so the average queue length at link *l* is**

$$N(l) = \frac{B(l)}{1-B(l)}, l \in \mathbf{L}$$

where

$$B(l) = \sum_{k \in \mathbf{U}} [q(l,k) + \sum_{i \in \mathbf{R}} c(l,(i,k))]$$

is the steady state probability that link *l* is busy

QoS metrics

- **The relevant QoS metrics, e.g. the total average delay through the network for a packet of class *k***

$$T(k) = \sum_{l \in \mathbf{L}} \pi(l,k) \frac{N(l)}{\Lambda_L(l,k)} + \sum_{r \in \mathbf{R}} \pi(r,k) \frac{N(r,k)}{\Lambda_R(l,k)}, \quad \bar{T} = \sum_k T(k)$$

where

$$\pi(r,k) = \frac{\Lambda_R(r,k)}{\lambda^+(k)}, r \in \mathbf{R} \qquad \pi(l,k) = \frac{\Lambda_L(l,k)}{\lambda^+(k)}, l \in \mathbf{L}$$

are the probabilities that a packet of class k enters router r or link l respectively, and the total traffic of class k, s being the source router of this class is

$$\lambda^+(k) = \sum\nolimits_{r \in \mathbf{R}} \lambda(r,k) = \lambda(s,k)$$

Imperial College London

Power Consumption Model

- **Routers**

$$P_i = \alpha_i + g_R(\Lambda_i) + c_i \sum_{k \in \mathbf{U}} \Lambda_R^-(i,(i,k)), i \in \mathbf{R}$$

where α_i is the static router power consumption, $g_R(.)$ is an increasing function of the packet processing rate as in Figure 1 and $c_i > 0$ is a proportionality constant related to the power consumed for the processing of the rerouting control

- **Links**

$$P_i = \beta_i + g_L(\Lambda_i), i \in \mathbf{L}$$

where β_i is the static power consumption when the link interface is on and $g_L(.)$ is an increasing function of the data transmission rate on the link as in Figure 2

Gradient Descent Optimisation

- The routing optimisation can be expressed as the minimization of a function that combines power consumption and (e.g.) the network average delay :

Minimize $G = c\sum_{i \in N} P_i + \overline{T}$

Using the $Q(i,k,j)$

- We therefore need to design algorithm to obtain the parameters $Q^o(i,k,j)$ at the operating points of the network

$$\underline{X} = [\underline{\lambda}, \underline{\lambda}^-, \underline{\mu}, \underline{P}^+, \underline{p}]$$

A. Gradient Descent Optimization

- **Algorithm of O(|*U*|.|*N*|^3) complexity [High!!]**
  - **Initialize the values *Q(i,k,j)* and choose *η*>0**
  - **Solve |*U*| systems of |*N* | non-linear equations to obtain the steady state probabilities *q(i,k) from G-network theory***
  - **Solve |*U*| systems of |*N* | linear equations for gradient descent using G-network theory**

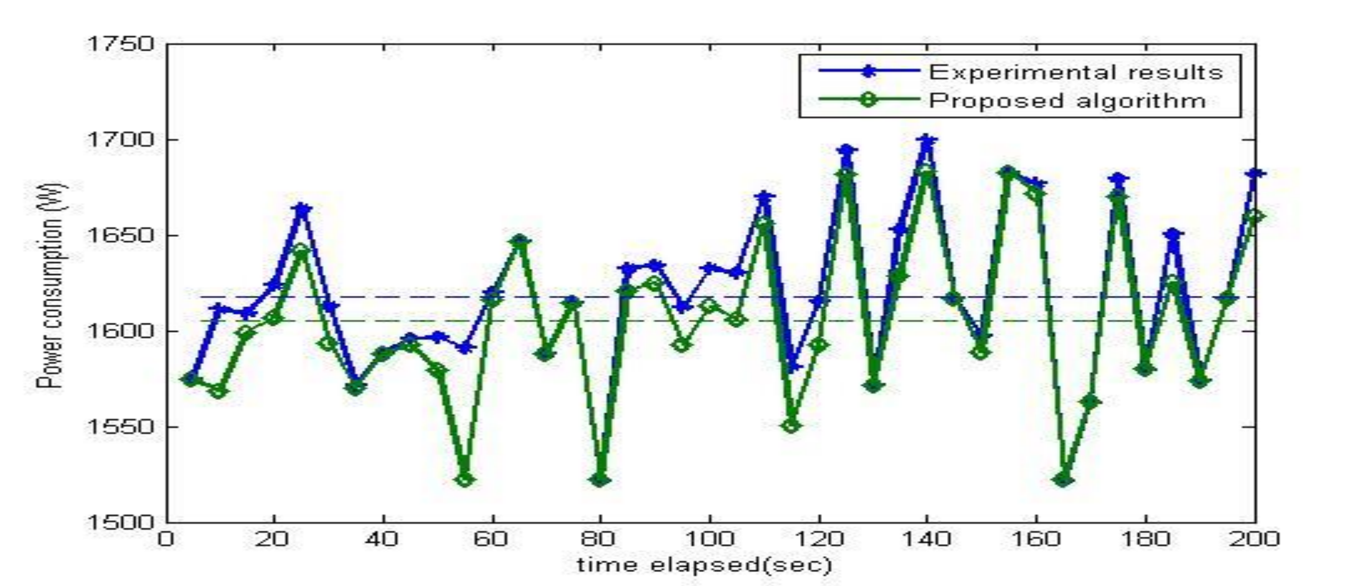
$$\frac{\partial \mathbf{q}_k}{\partial Q(x,m,y)} = \boldsymbol{\gamma}_k^{xmy}(\mathbf{I}-\mathbf{W}_k)^{-1}$$

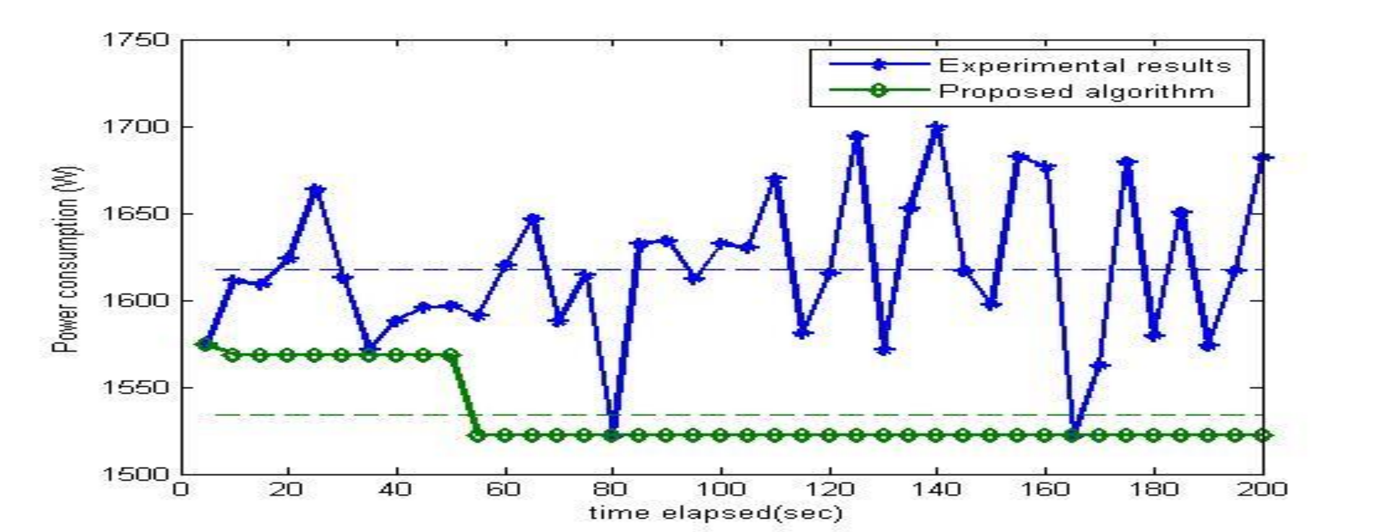
  - **Update the values of *Q(i,k,j)* using the nth computational step**

$$Q_{n+1}(i,k,j) = Q_n(i,k,j) - \eta \frac{\partial G}{\partial Q(i,k,j)}\Big|_{Q(i,k,j)=Q_n(i,k,j)}$$

Imperial College London

Gradient Descent on Top of EARP





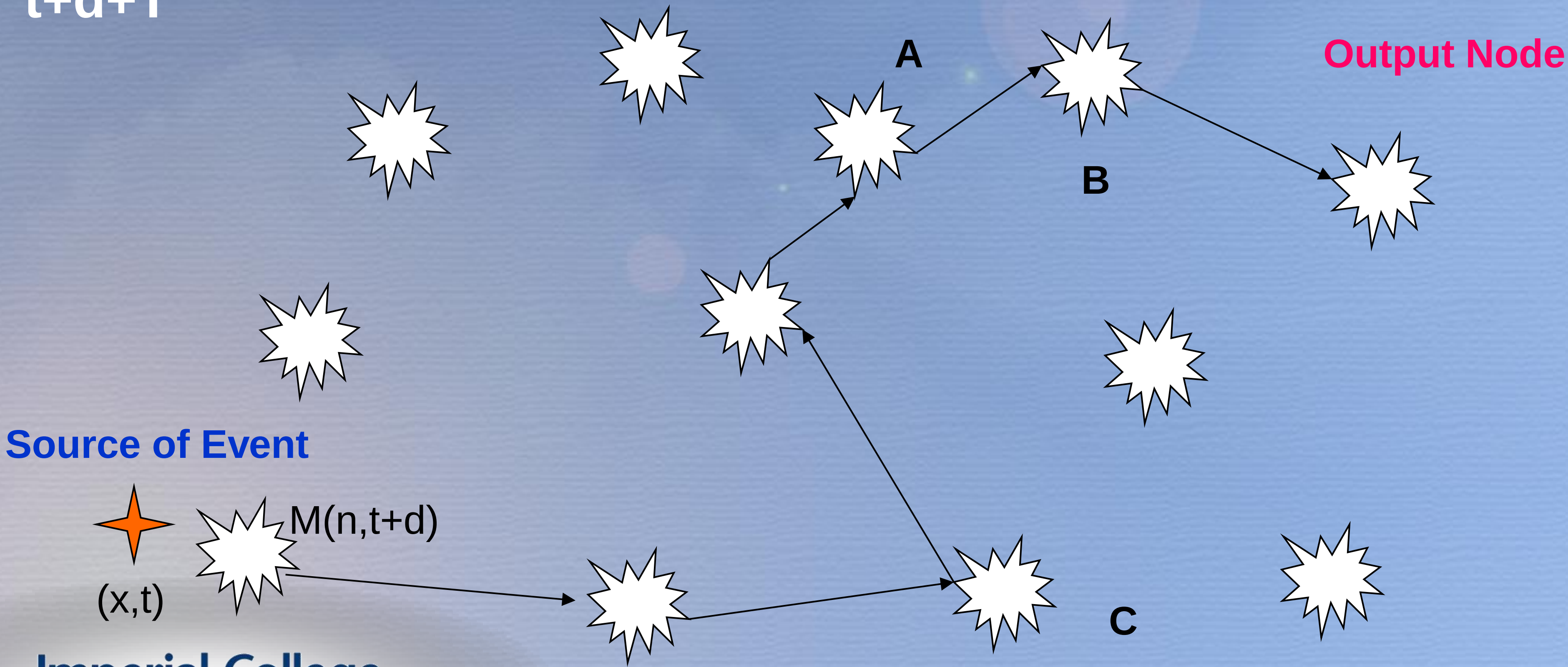
Imperial College London

A Model for Time & Energy that is both Cyber & Physical, and E. Gelenbe Phys Rev Dec 2010

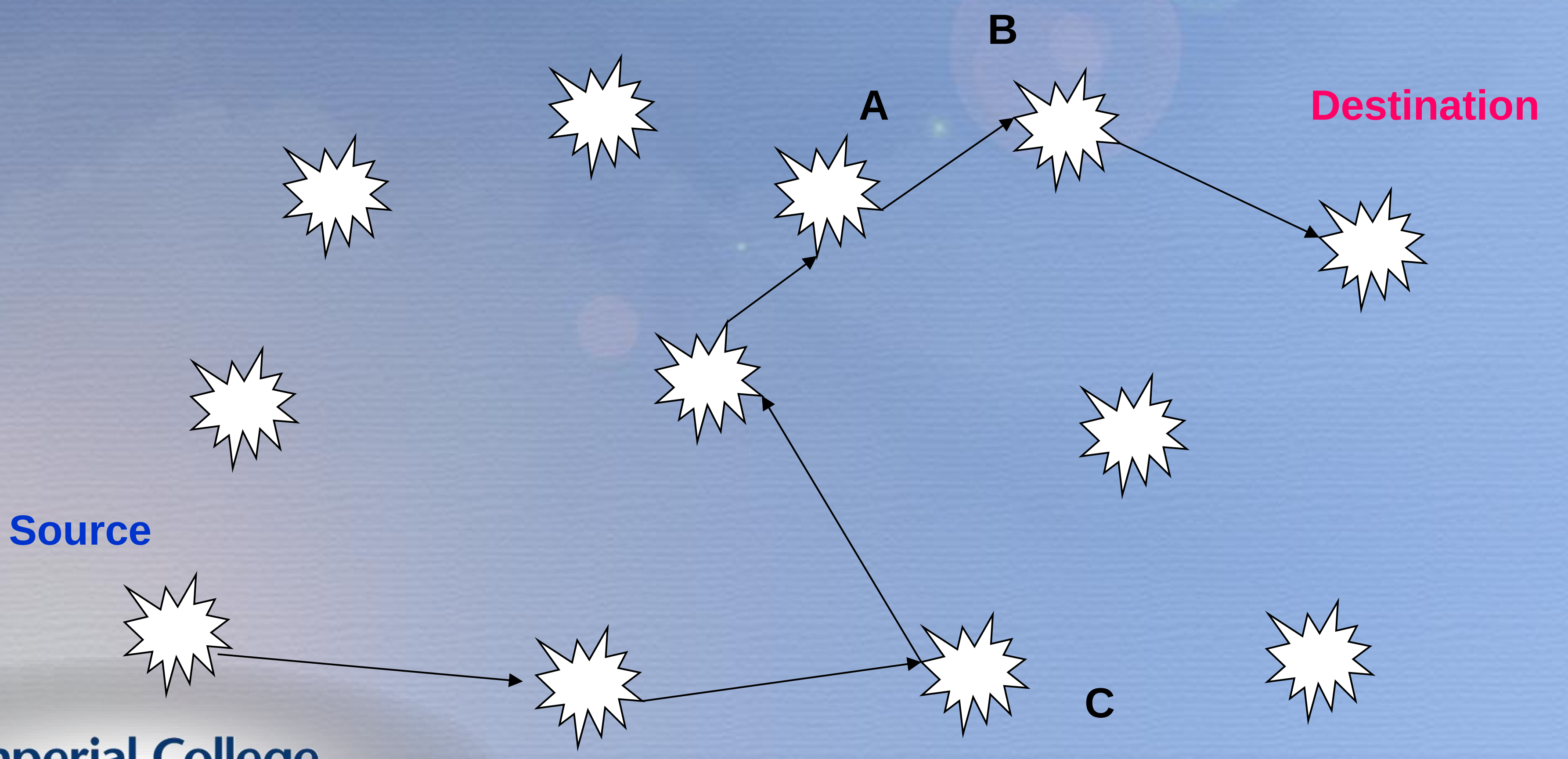
- N robots or people Search in an Unknown & Large City
- N Packets Travel in a Very-Large Network
- Search by Software Robots for Data in a Very Large Distributed Database
- Biological Agents Diffusing through a Random Medium until they Encounter a Docking Point
- Particles Moving in a Random Medium until they Encounter an Oppositely Charged Receptor
- Randomised Gradient Minimisation (e.g. Simulated Annealing) on Parallel Processors

Example from Wireless Sensor Networks

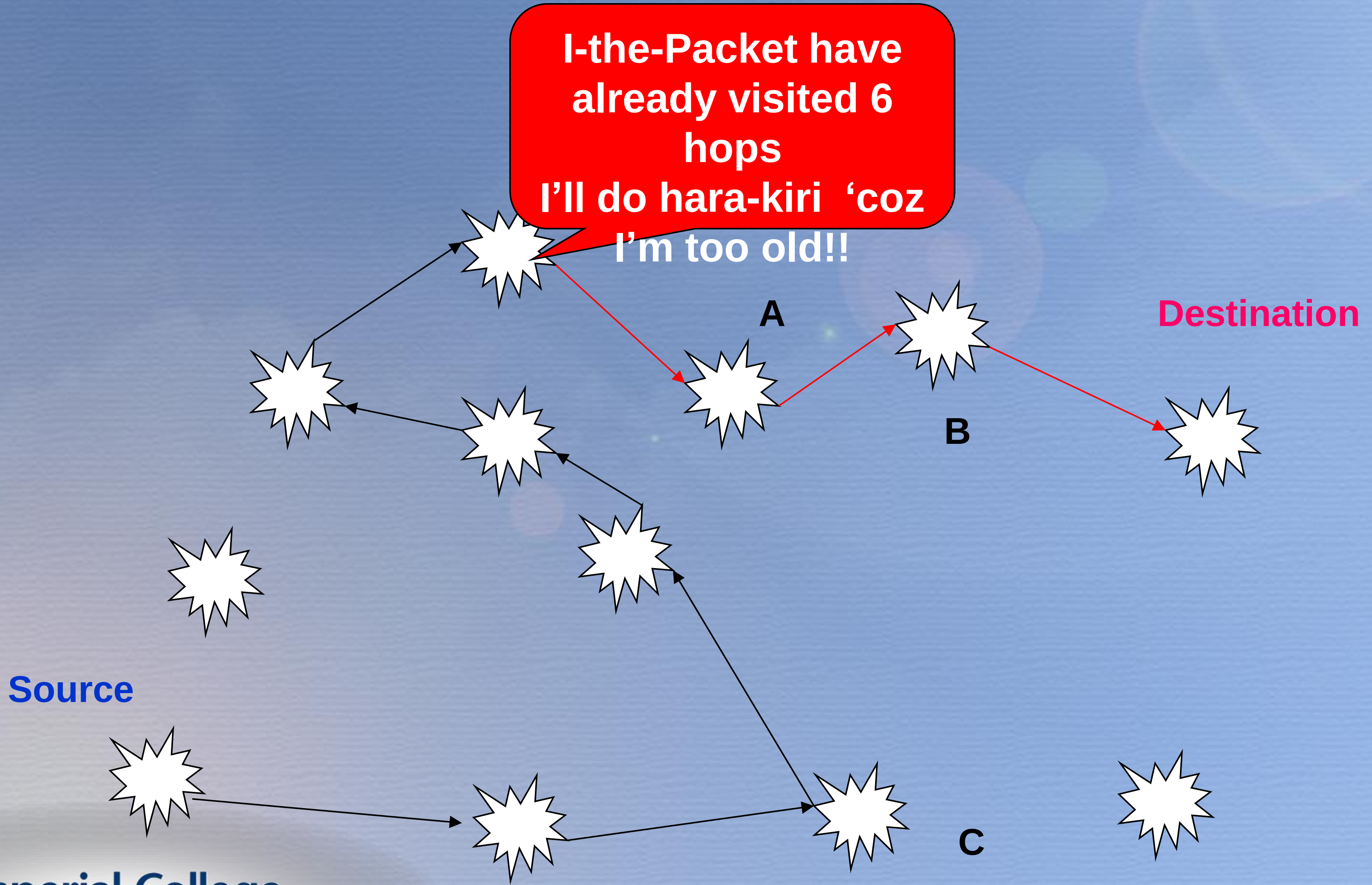
Event occuring at location (x,t) is reported by the Sensor Node at location (n,t+d) if $||X(n)-x||<\varepsilon$. The node sends out a packet at t+d. The packet containing M(n,X(n),t+d) travels over multiple hops and reaches the Output Node at time t+d+T



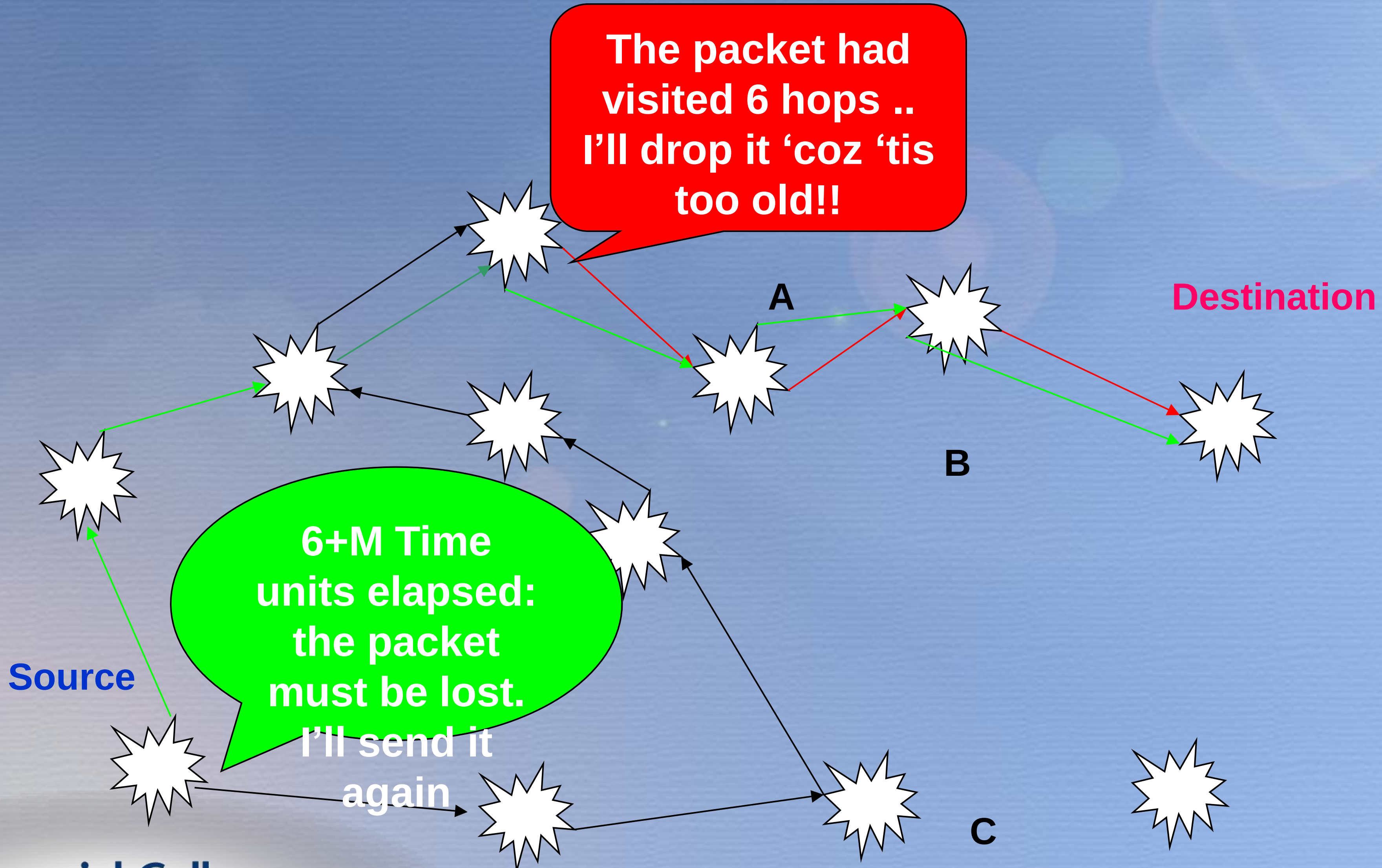
A Packet Needs to Go From S to Destination Using Multiple Hops .. But it is Ignorant about its Path and all Kinds of Bad Things Can Happen .. Can it Still Succeed?



Yet Another Situation .. Packet Hara Kiri



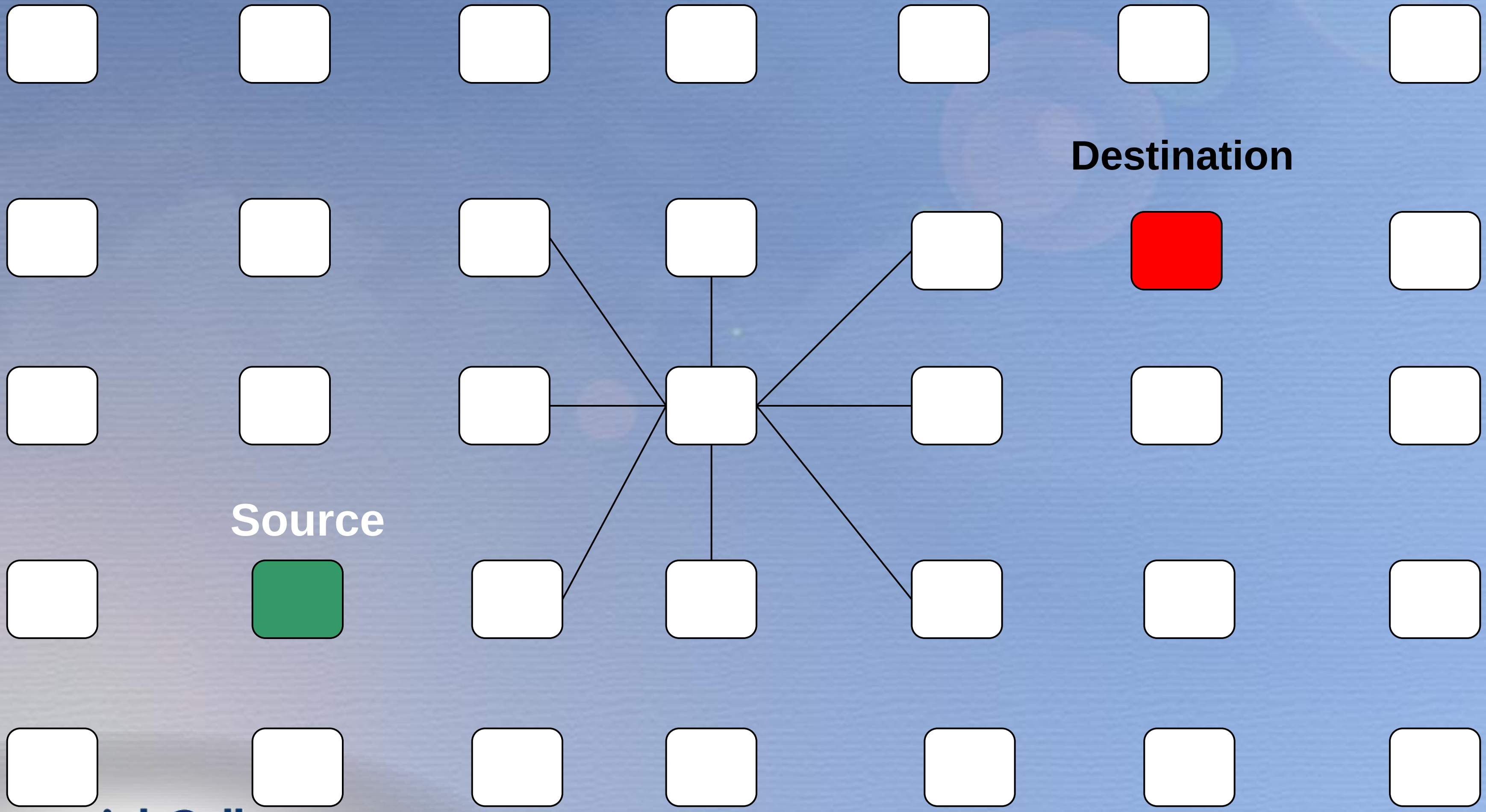
Some Time Later .. Packet Retransmission



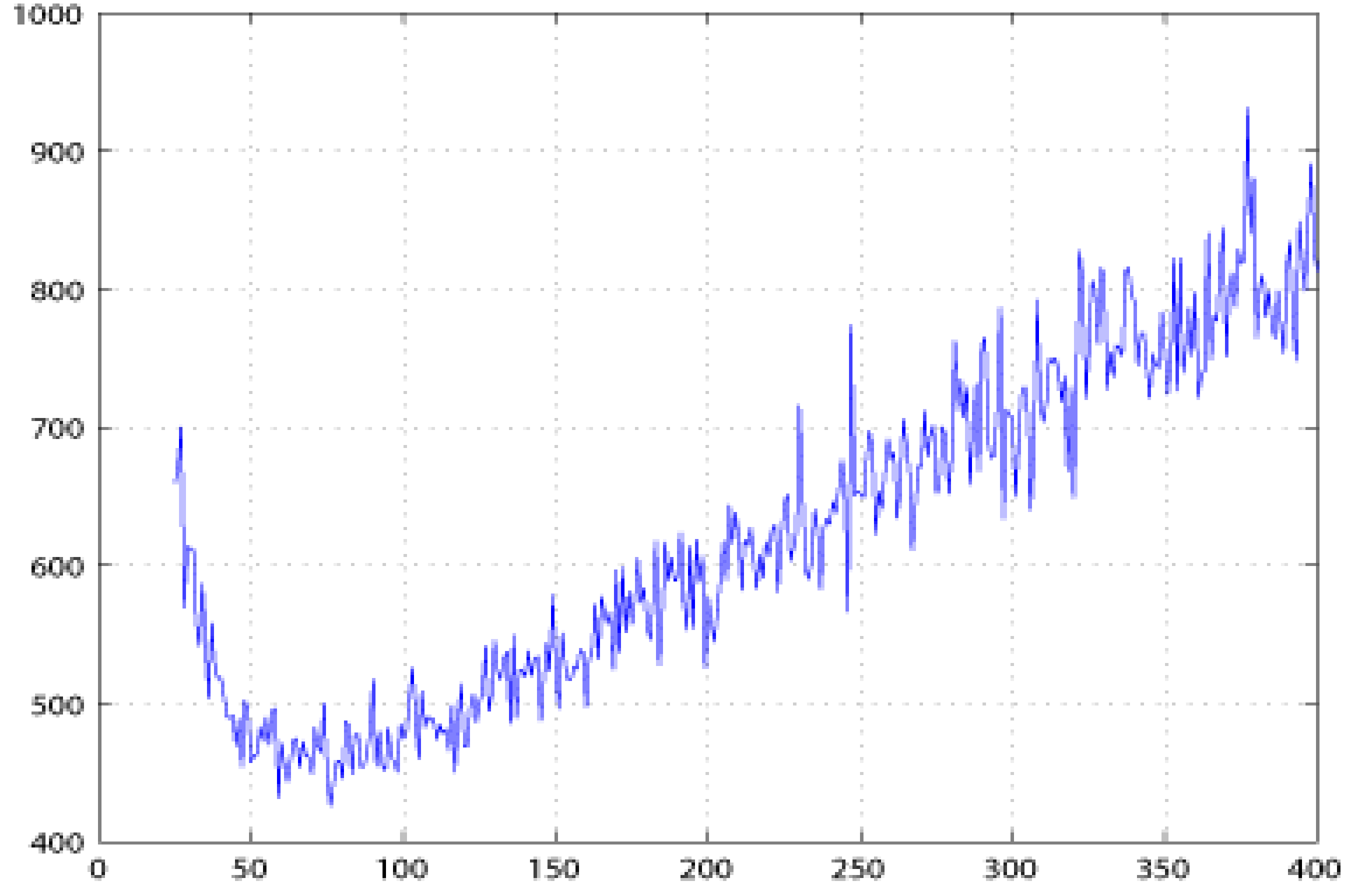
Network Model

- Packets go from some source S to a Destination (that may move) that is initially at distance D
- The wireless range is $\delta << D$, there are no collisions
- Packets can be lost in $[t,t+\Delta t]$ with probability $\lambda\Delta t$ anywhere on the path
- There is a time-out R (in time or number of hops), modelled as being timed-out in $[t,t+\Delta t]$ with probability $r\Delta t$ with a subsequent retransmission delay M
- Packets may or not know the direction they need to go – we do not nail down the routing scheme with any specific assumptions
- We avoid assumptions about the geography of nodes in m-dimensions, and assume temporal and spatial homogeneity and temporal and spatial independence

Simulation examples in an infinite grid



Simulations of Average Travel Time vs Constant Time-Out
δ=1, D=10, M=20, No Loss
Perfect Ignorance: b=0, c=1



Diffusion Model

- Do not consider the detailed topology of nodes,

- Assume homogeneity with respect to the distance to destination, and over time,

- Represent motion as a continuous process, for packets it would be a continuous approximation of discrete motion,

- Allow for loss (of packets) or destruction of the robotic searcher, or inactivation of the biological agent

- Include a time-out for the source to re-send the packet

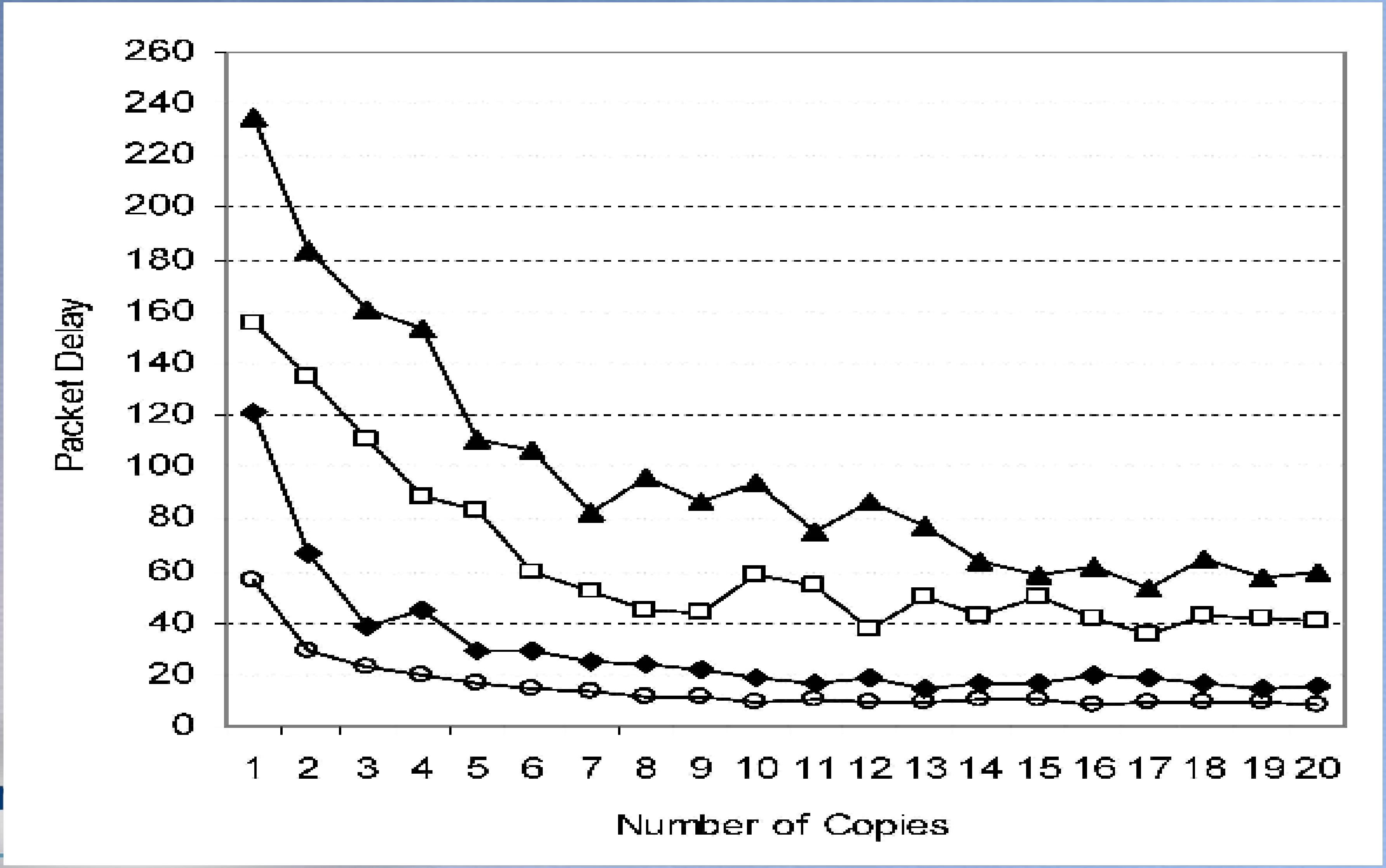
- After each Time-Out, the sender waits M time units and then retransmits the packet under identical statistical conditions

- The distance of the searcher with respect to the destination at time t is X(t); it is homogeneous with respect to position and time
- Motion of the searcher is characterised by parameters b and c
    - The drift $b = E[X(t+\Delta t) - X(t)|X=x] / \Delta t$
    - The instantaneous variance
      $$c = E[(X(t+\Delta t) - X(t) - b\Delta t)^2|X=x] / (\Delta t)^2$$

- Loss (of packets), destruction of the robotic searcher, inactivation of the biological agent , represented by $\lambda\Delta t$

- Time-out is represented by $r\Delta t$, and after each Time-out, the sender waits M (on average $1/\mu$) time units and then resends the packet which then travels under iid statistical conditions

N independent searchers: find average time for the first one to get there



$$\frac{\partial f_i}{\partial t} = -b\frac{\partial f_i}{\partial x_i} + \frac{1}{2}c\frac{\partial^2 f_i}{\partial {x_i}^2} - a_i f_i + [\mu W_i(t) + P_i(t)]\delta(x_i - D)$$

$$\frac{dP_i(t)}{dt} = -P_i(t) + \sum_{i=1}^{N} \lim_{x_i \to 0^+} [-bf_i + \frac{1}{2}c\frac{\partial f_i}{\partial x_i}]$$

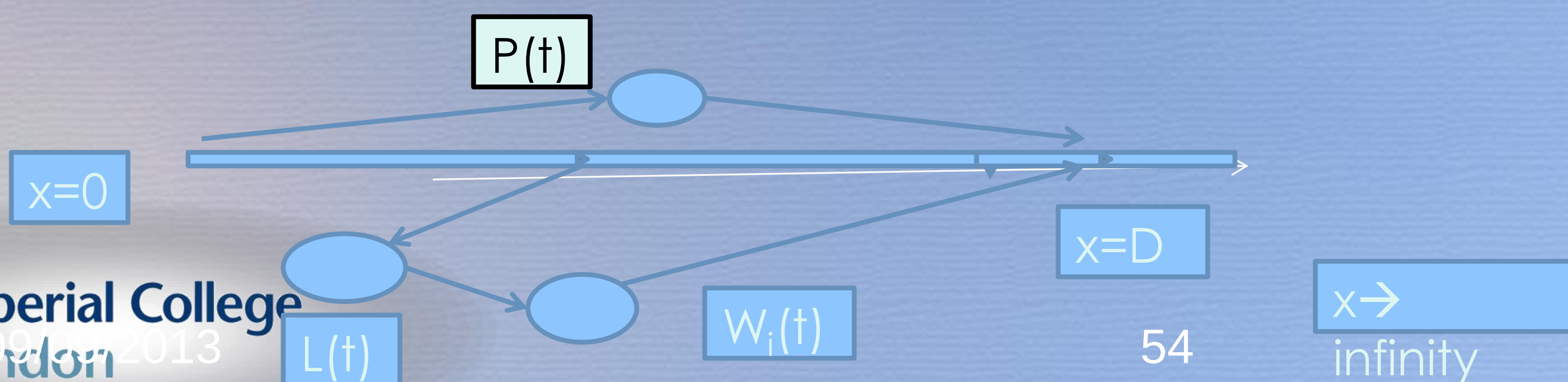
$$\frac{dL_i(t)}{dt} = \lambda \int_{0^+}^{\infty} f_i dx_i - (r + a_i)L_i(t)$$

$$\frac{dW_i(t)}{dt} = r\int_{0^+}^{\infty} f_i dx_i + rL_i(t) - (\mu + a_i)W_i(t)$$

$$a_j = -\sum_{i=1, i \neq j}^{N} \lim_{x_i \to 0^+} [-bf_i + \frac{1}{2}c\frac{\partial f_i}{\partial x_i}],$$

$$P_i(t) + L_i(t) + W_i(t) + \int_{0^+}^{\infty} f_i dx_i = 1; \quad \lim_{x \to 0^+} f = 0.$$

$$\mathrm{E[T^*]} = \mathrm{P_i}^{-1} - 1 \text{ obtained from the stationary solution}$$

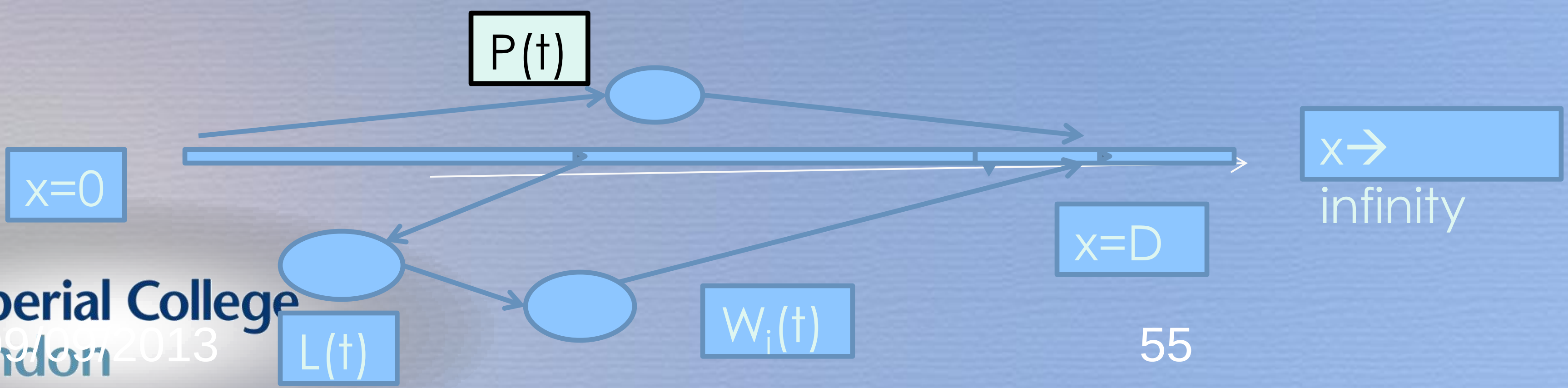


$$f_i(z) = A[e^{u_1 z} - e^{u_2 z}], 0 \le z \le D$$

$$f_i(z) = A[e^{(u_1 - u_2)D} - 1]e^{u_2 z}, z \ge D$$

$$u_{1,2} = \frac{b \pm \sqrt{+2c(\lambda + r + a)}}{c}$$

$$a_i = \sum_{j=1, i \neq j}^{N} \lim_{z_j \to 0} [b f_j(z_j) + \frac{1}{2} c \frac{\partial^2 f_j(z_j)}{\partial z_j^{\ 2}}]$$



Expected Travel Time to Destination for N Searchers with initial Distance D

$T^* = \inf\{T_1, \dots, T_N\}$

- Drift $b \leq 0$ or $b>0$, Second Moment Param. $c \geq 0$
- Avg Time-Out $R=1/r$, $M=1/\mu$, then we derive:

$$E[T \mid D] = \frac{1}{N}[e^{-2D(\frac{\lambda+r+a}{b-\sqrt{b^2+2c(\lambda+r+a)}})} - 1][\frac{\mu+r+a}{(r+a)\mu+a}]$$

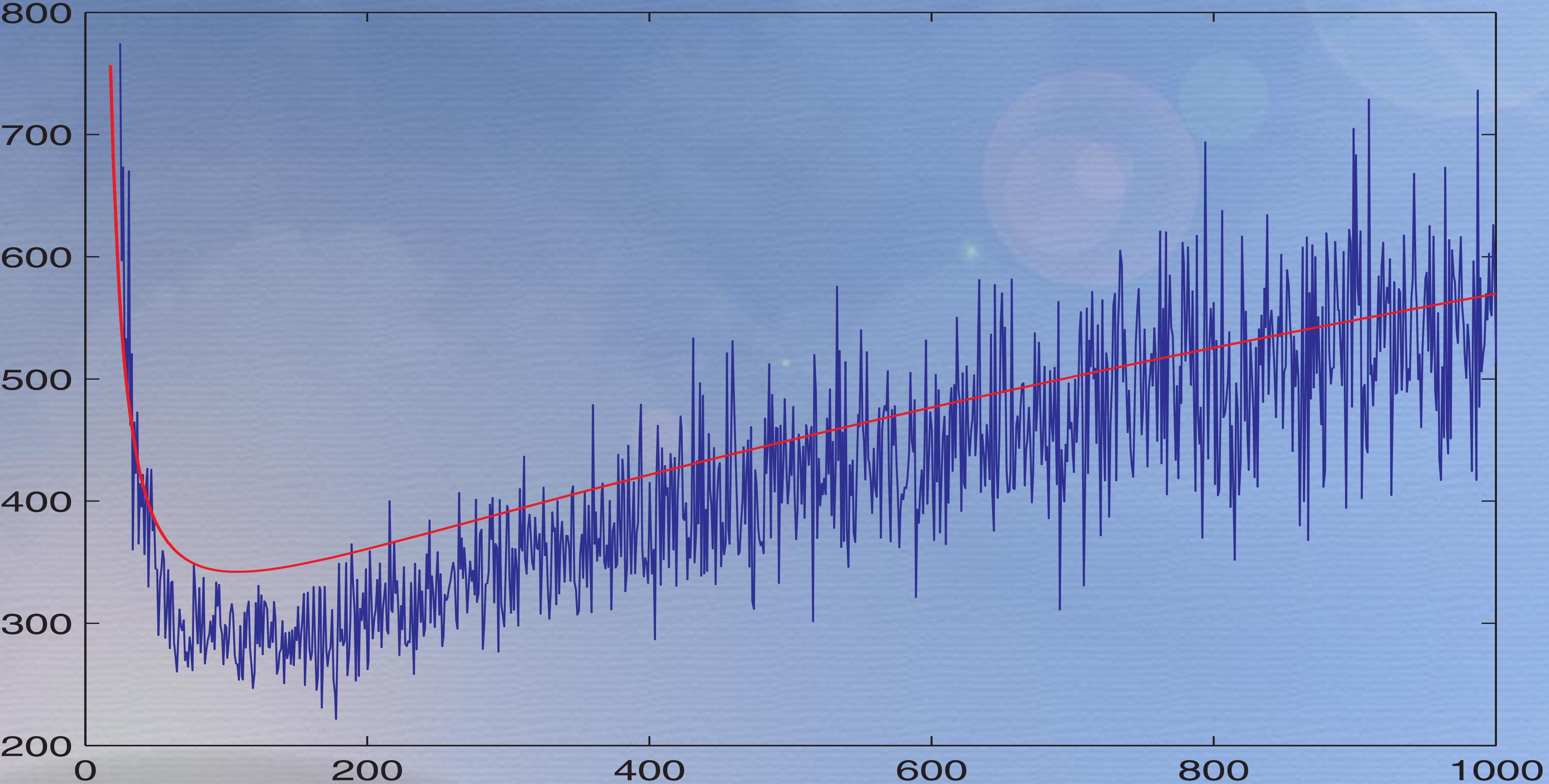
Effective Travel Time & Energy

$$T^* = \inf\{T_1, \dots, T_N\}$$

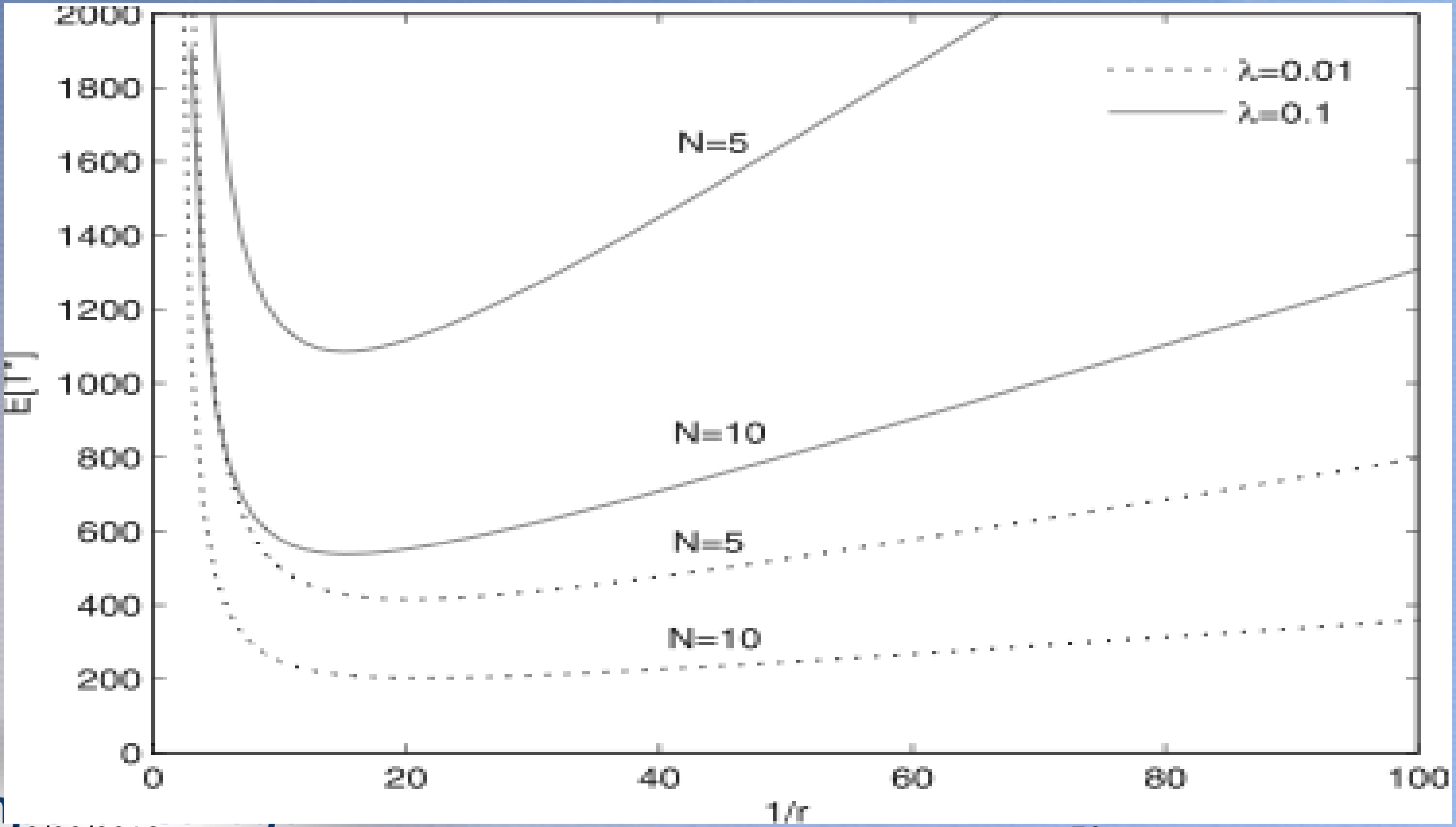
- $E[\tau_{eff} \mid D] = [1+E[T^* \mid D]\,]\,.\,P[\text{searcher is moving}]$
- $J(N \mid D) = N.E[\tau_{eff} \mid D]$

$$J(N \mid D) = [e^{-2D\left(\frac{\lambda+r+a}{b-\sqrt{b^2+2c(\lambda+r+a)}}\right)} - 1][\frac{1}{\lambda+r+a}]$$

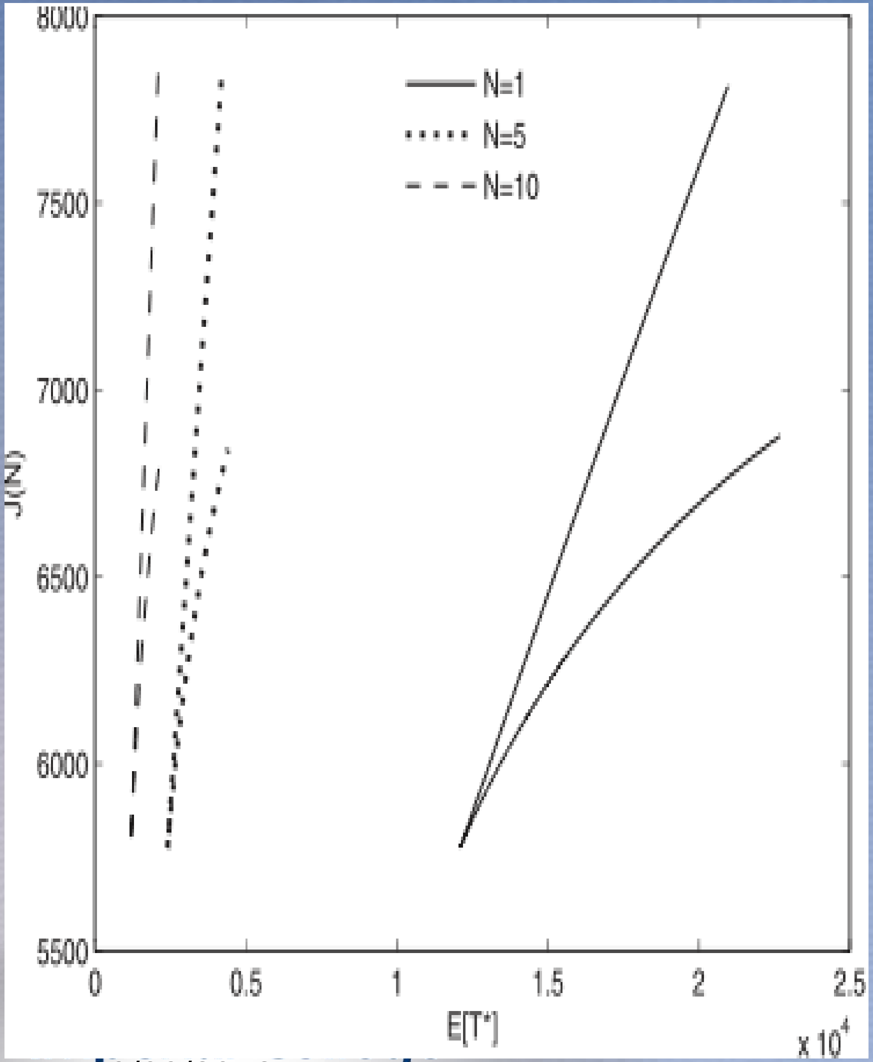
Comparing Theory with Simulation for N=1

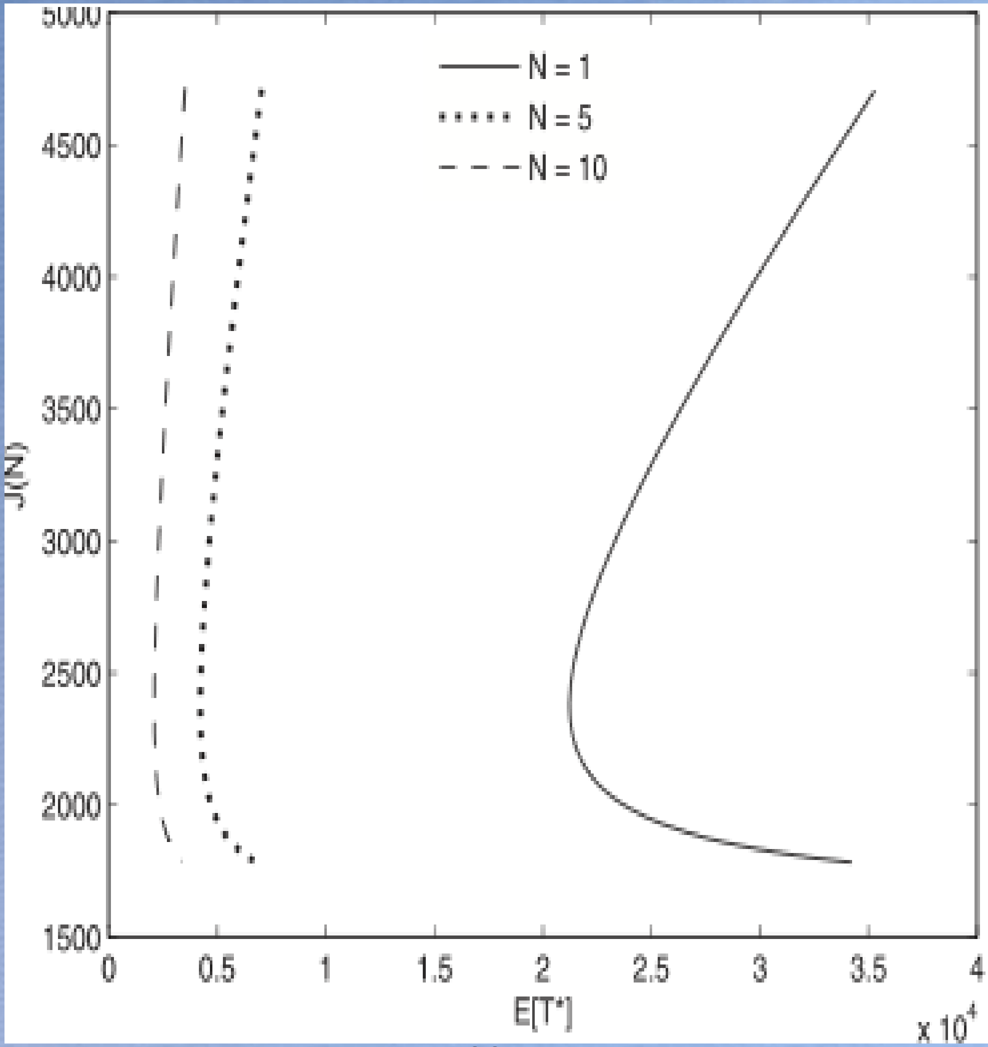


Average Travel Time vs Time-Out and Different N and Loss Rates

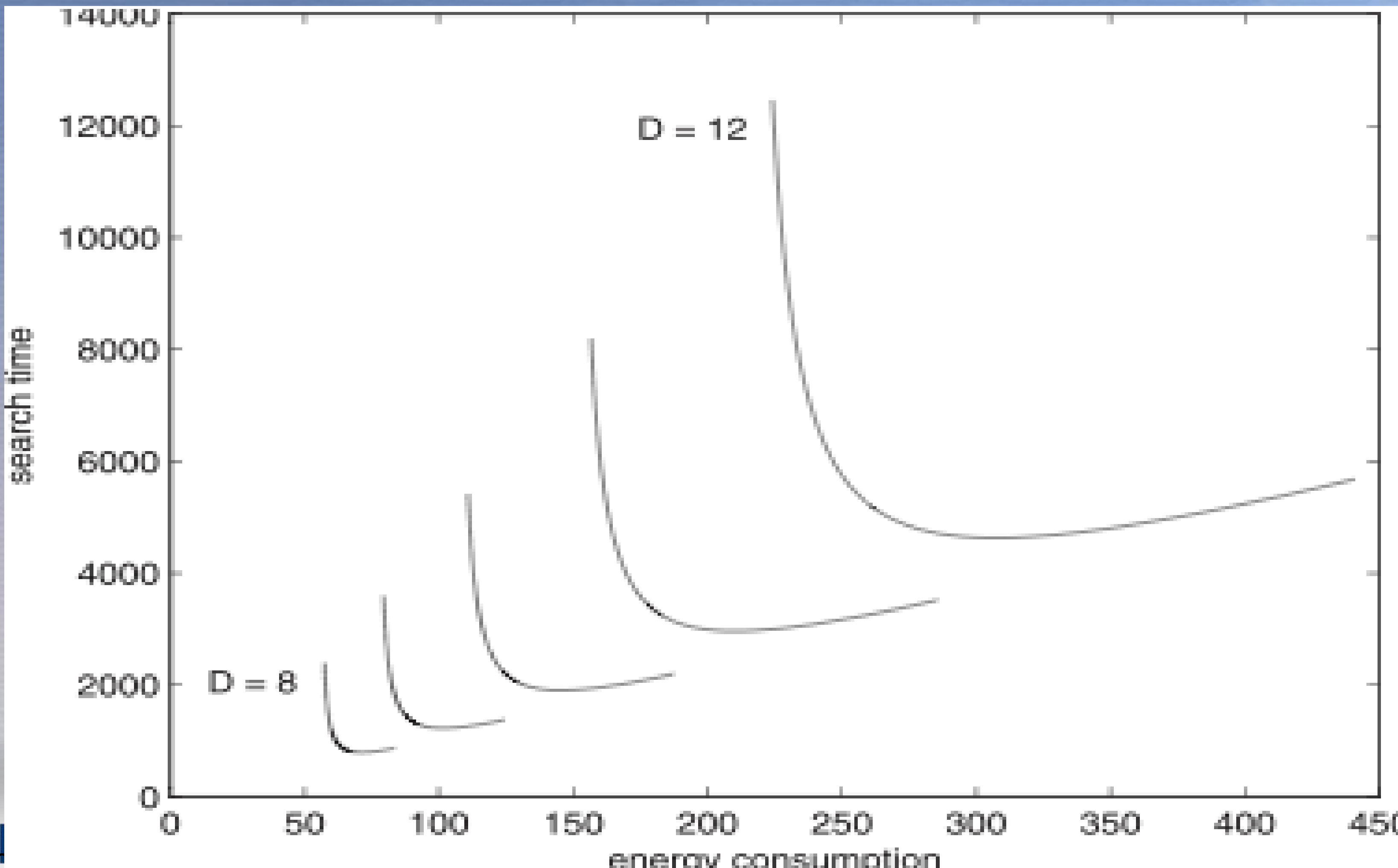


Locus of Average Time and Energy vs Time-Out





Locus of Average Time and Energy vs Time-Out with Different Distances



Single packet travel delay in a wireless network with imperfect routing and packet losses

- **E. Gelenbe "A Diffusion model for packet travel time in a random multi-hop medium", ACM Trans. on Sensor Networks, Vol. 3 (2), p. 111, 2007**

N Packets or Searchers sent simultaneously in a homogenous environment:

- **E. Gelenbe "Search in unknown random environments", Physical Review E82: 061112 (2010), Dec. 7, 2010.**

Results ACM MAMA 2011

Omer Abdelrahman & Erol Gelenbe

Single packet travel delay in a wireless network with non-homogenous parameters, imperfect routing and packet losses

- **Large Network with Non-Homogenous Coverage**
- **Modeling an Attacking Packet in the presence of Defense Near the Target (Destination) Node**

→ Phase Transition Effect

Non-Homogenous Case

Original

$$\frac{\partial f(z,t)}{\partial t} = \frac{1}{2}\frac{\partial^2[c(z)f(z,t)]}{\partial z^2} - \frac{\partial[b(z)f(z,t)]}{\partial z} - (\lambda(z)+r)f(z,t) + [P(t)+\mu W(t)]\delta(z-D)$$

$$\frac{dL(t)}{dt} = -rL(t) + \int_0^\infty \lambda(z)f(z,t)dz$$

$$\frac{dW(t)}{dt} = -\mu W(t) + r[L(t) + \int_0^\infty f(z,t)dz]$$

$$\frac{dP(t)}{dt} = -P(t) + \lim_{z\to 0^+}[\frac{1}{2}\frac{\partial[c(z)f(z,t)]}{\partial z} - b(z)f(z,t)]$$

$$1 = P(t) + W(t) + L(t) + \int_0^\infty f(z,t)dz$$

Discretized

$$0 = \frac{c_k}{2}\frac{d^2 f_k(z)}{dz^2} - b_k\frac{df_k(z)}{dz} - (\lambda_k + r)f_k(z) \qquad (1)$$

while the equation for the segment where the source is located is:

$$-[P+\mu W]\delta(z-D) = \frac{c_n}{2}\frac{d^2 f_n(z)}{dz^2} - b_n\frac{df_n(z)}{dz} - (\lambda_n + r)f_n(z) \qquad (2)$$

We will also have:

$$rL = \sum_{k=1}^{m}\lambda_k\int_{Z_{k-1}}^{Z_k} f_k(z)dz \qquad (3)$$

$$\mu W = r[L + \sum_{k=1}^{m}\int_{Z_{k-1}}^{Z_k} f_k(z)dz] \qquad (4)$$

$$P = \lim_{z\to 0^+}[\frac{c_1}{2}\frac{df_1(z)}{dz} - b_1 f_1(z)] \qquad (5)$$

and the normalization condition:

$$1 = P + W + L + \sum_{k=1}^{m}\int_{Z_{k-1}}^{Z_k} f_k(z)dz \qquad (6)$$

Discretized Segments

$$E[T] = \left(\frac{1}{r} + \frac{1}{\mu}\right) \times$$
$$\left[\sqrt{\frac{b_n^2 + 2c_n(\lambda_n + r)}{b_1^2 + 2c_1(\lambda_1 + r)}}\ \frac{\overline{A}_n\overline{G}_n e^{u_n S_n} - \overline{B}_n\overline{F}_n e^{v_n S_n}}{\overline{G}_n e^{u_n(Z_n - D)} + \overline{F}_n e^{v_n(Z_n - D)}} - 1\right] \tag{7}$$

where the remaining parameters are computed as follows. Define:

$$\alpha_k^- = \frac{c_k u_k - c_{k-1} v_{k-1}}{c_k(u_k - v_k)}, \qquad \beta_k^- = \frac{c_k u_k - c_{k-1} u_{k-1}}{c_k(u_k - v_k)}$$
$$\alpha_k^+ = \frac{c_k u_k - c_{k+1} v_{k+1}}{c_k(u_k - v_k)}, \qquad \beta_k^+ = \frac{c_k u_k - c_{k+1} u_{k+1}}{c_k(u_k - v_k)} \tag{8}$$

Then set $\overline{A}_1 = 1$ and $\overline{B}_1 = -1$ and for $2 \leq k \leq n$ compute:

$$\begin{bmatrix} \overline{A}_k \\ \overline{B}_k \end{bmatrix} = \begin{bmatrix} \alpha_k^- & \beta_k^- \\ 1-\alpha_k^- & 1-\beta_k^- \end{bmatrix} \begin{bmatrix} e^{u_{k-1}S_{k-1}} & 0 \\ 0 & e^{v_{k-1}S_{k-1}} \end{bmatrix} \begin{bmatrix} \overline{A}_{k-1} \\ \overline{B}_{k-1} \end{bmatrix} \tag{9}$$

Then set $\overline{F}_m = 0$ and $\overline{G}_m = e^{v_m Z_m}$, and start another computation at $k = m-1$ for $n \leq k \leq m-1$ with:

$$\begin{bmatrix} \overline{F}_k \\ \overline{G}_k \end{bmatrix} = \begin{bmatrix} \alpha_k^+ & \beta_k^+ \\ 1-\alpha_k^+ & 1-\beta_k^+ \end{bmatrix} \begin{bmatrix} e^{-u_{k+1}S_{k+1}} & 0 \\ 0 & e^{-v_{k+1}S_{k+1}} \end{bmatrix} \begin{bmatrix} \overline{F}_{k+1} \\ \overline{G}_{k+1} \end{bmatrix}$$

Discretized Segments

Remark 1 With n being the index of the discretisation segment that includes the source node at D, it is interesting to see that $E[T]$ only depends on a set of parameters that are computed for values of $k = 1$, $k = n$, and on two sets of algebraic iterations between $k = 1$ and $k = n$ and $k = m$ down to $k = n$.

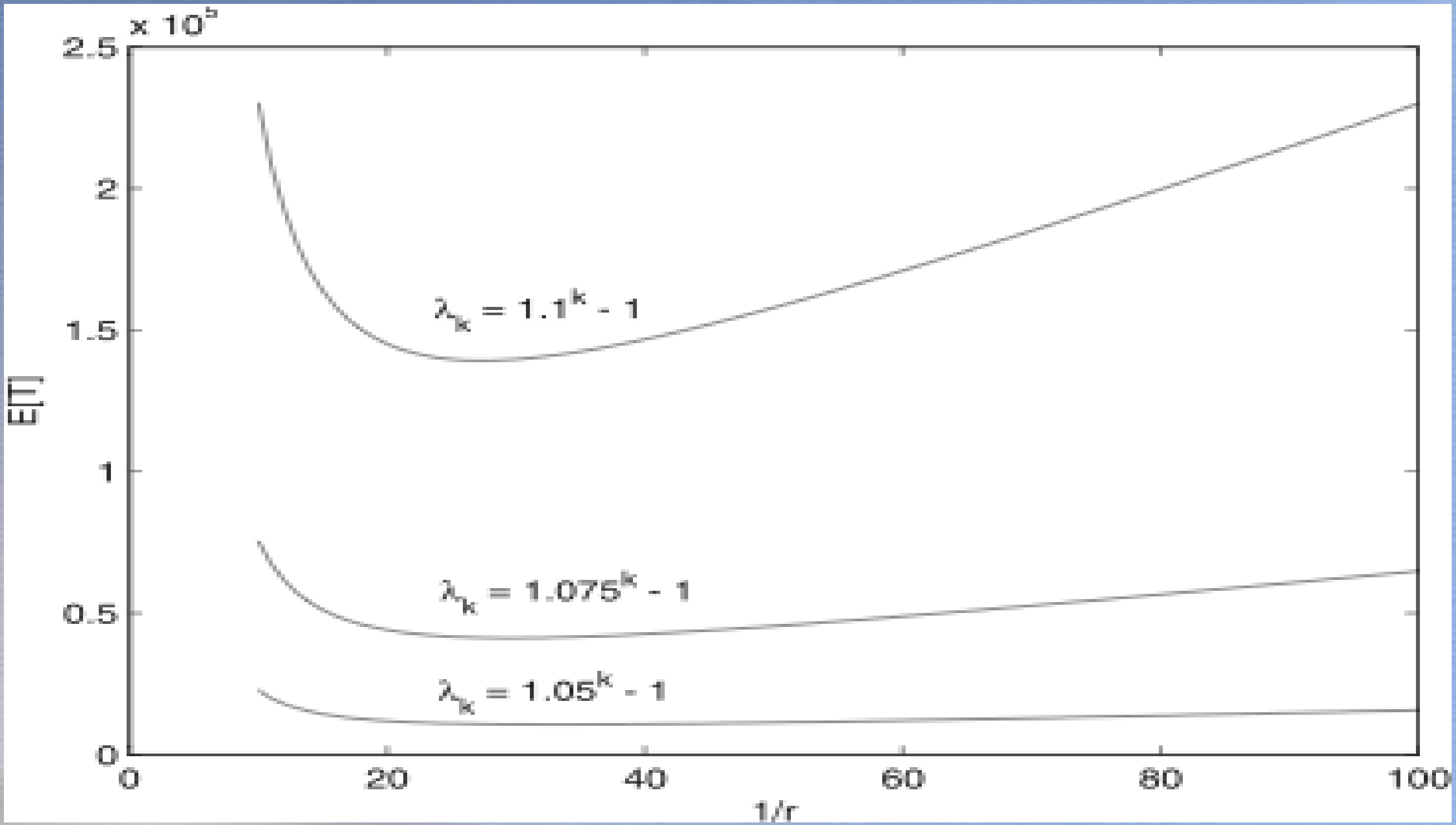
Remark 2 When the source node is in the pen-ultimate segment we have $m = n$, and:

$$E[T] = \frac{r+\mu}{r\mu}\left[\sqrt{\frac{b_n^2 + 2c_n(\lambda_n + r)}{b_1^2 + 2c_1(\lambda_1 + r)}}\,\overline{A}_n e^{u_n(D - Z_{n-1})} - 1\right] \quad (20)$$

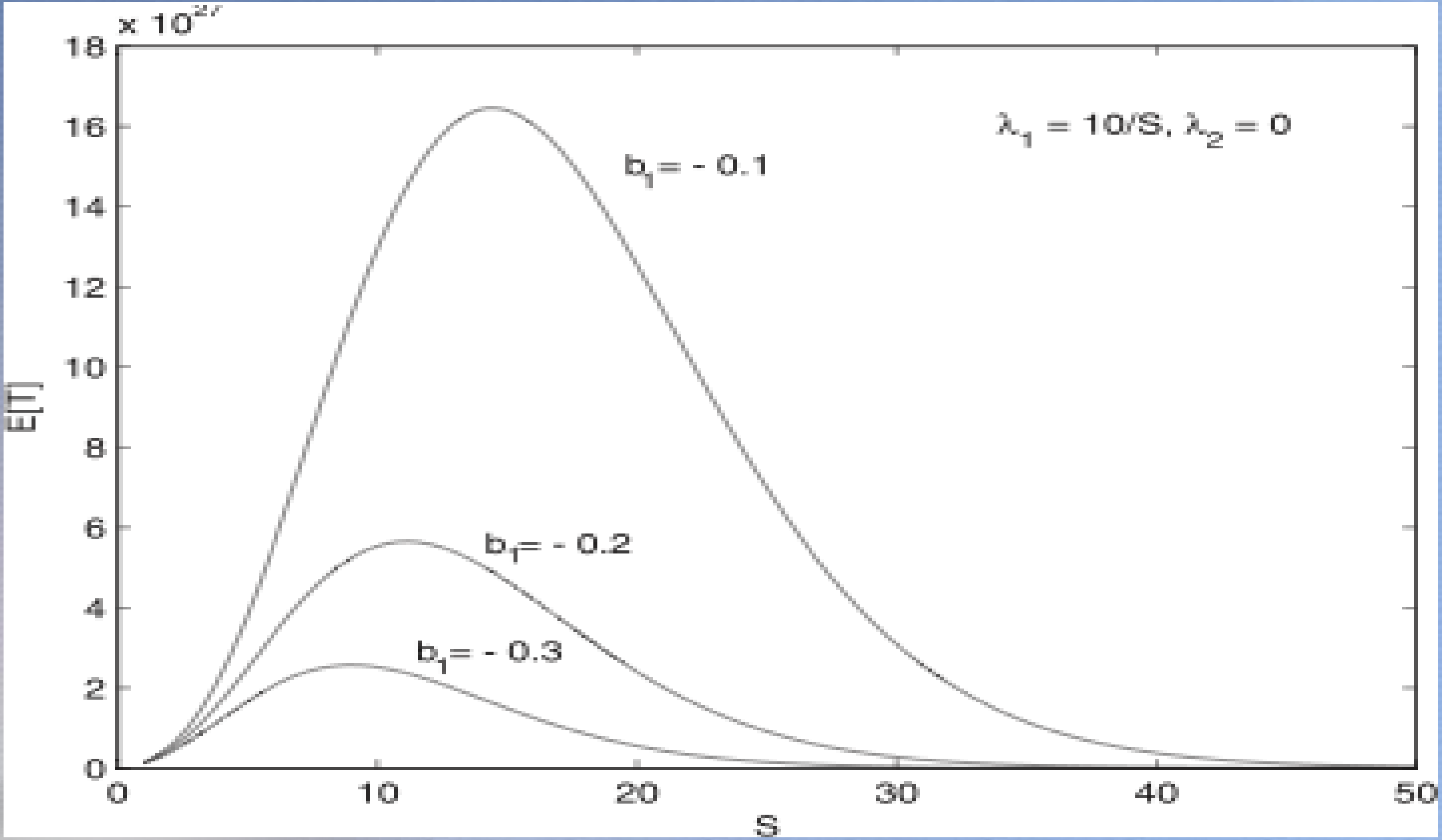
For a homogenous medium $m = n = 1$ and:

$$E[T] = \left(\frac{1}{r} + \frac{1}{\mu}\right)\left[e^{u_1 D} - 1\right]$$

Increased Drop Rate Near the Destination Makes it Harder to Reach the Destination



Protected Area of Size S Around Destination with Intrusion Detection and Drops



Protected Destination with Perfect Routers b= -1

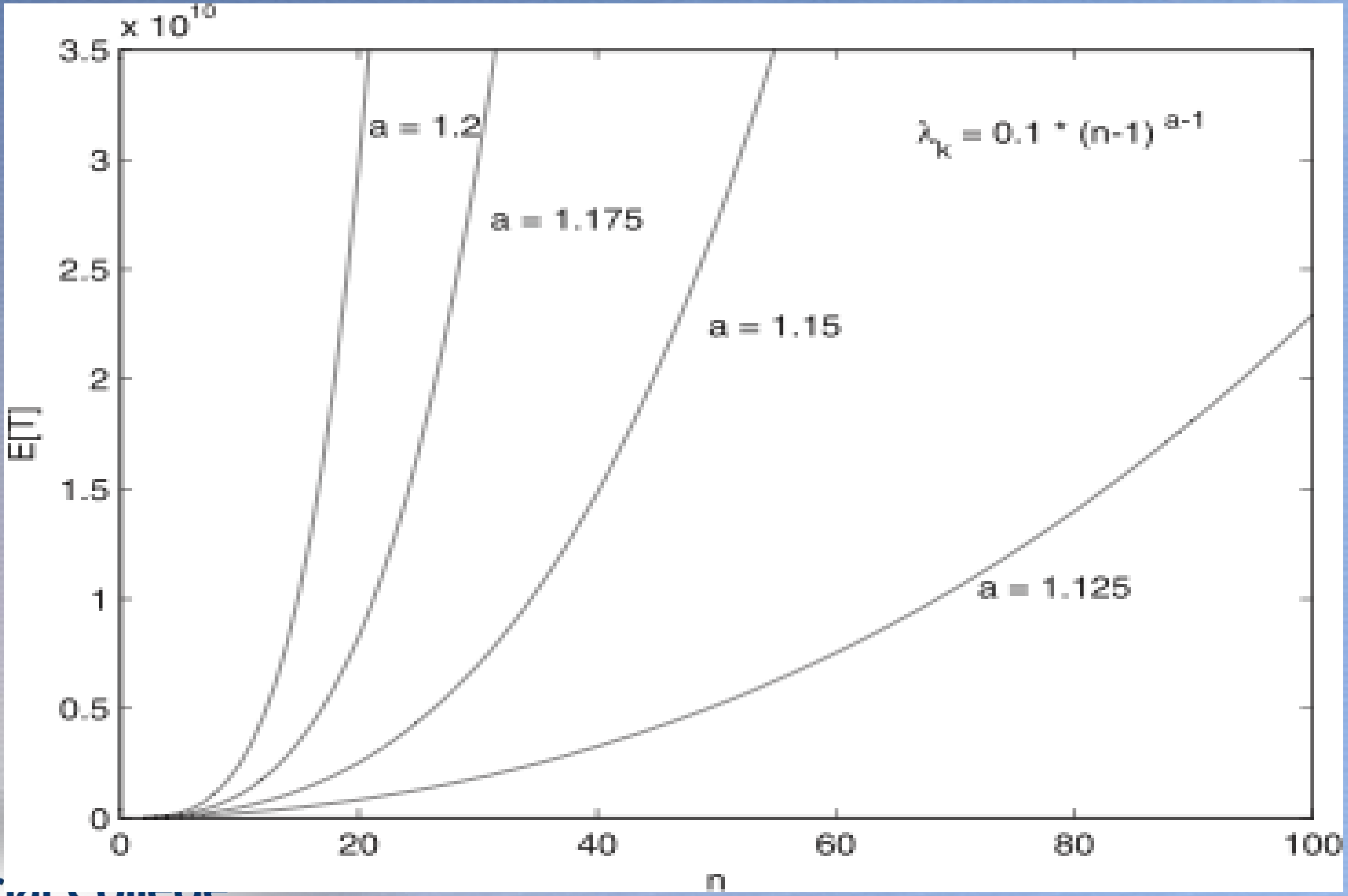
Now let us introduce a non-homogenous packet drop effect by choosing an integer n to create an acceleration in the packet drop effect and let $S_i = D/(n-1)$ so that:

$$E[T] = \frac{r+\mu}{r\mu}\,[e^{rD}\;e^{D\frac{\sum_{i=1}^{n-1}\lambda_i}{n-1}} - 1] \qquad (24)$$

which yields the following result.

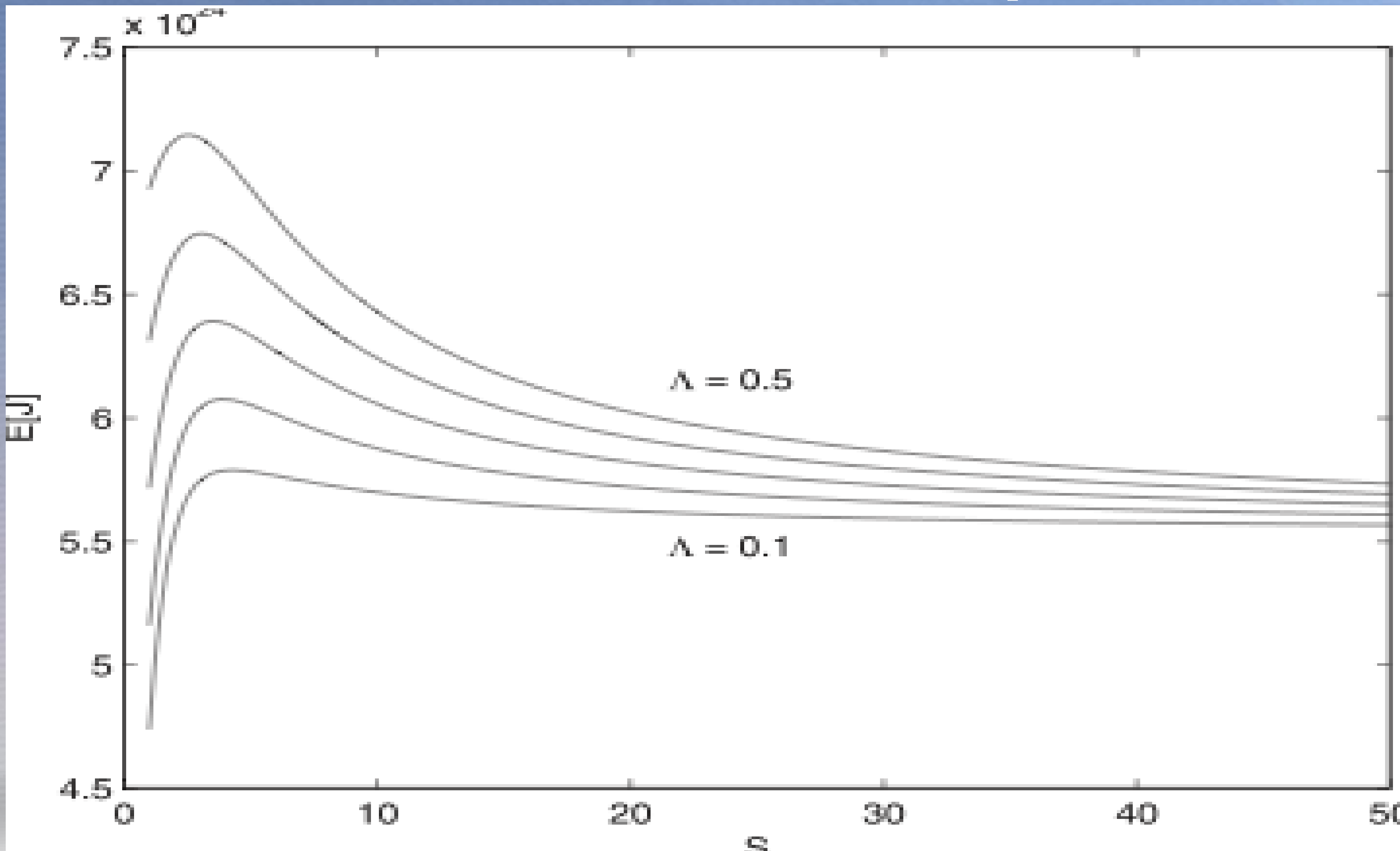
Result 4 If $\lim_{n\to\infty} \frac{\sum_{i=1}^{n-1}\lambda_i}{n-1} = +\infty$ then the packet will never reach the destination node. Otherwise it will reach it in a time which is finite on average, and with probability one. The Figure 2 illustrates **Result 4** by showing that even with a small excess, represented by $a > 1$, above the $o(n)$ rate of increase for the loss rate λ_k the attacking packet's progress will be indefinitely impeded by the drops, despite the subsequent time-outs.

Increased Drop Rate Near the Destination Makes it Hard to Reach the Destination

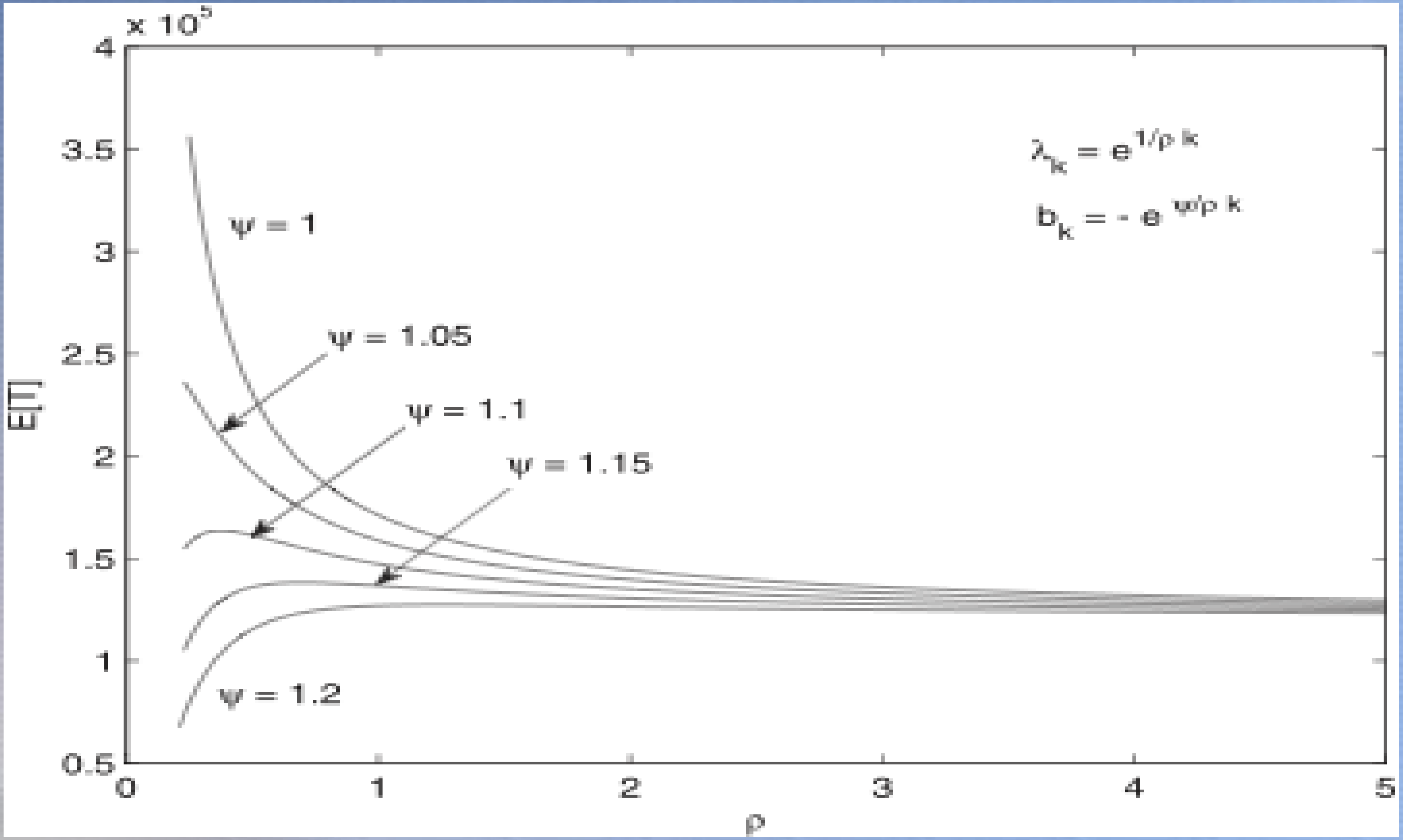


Imperial College London

Energy Consumption: Protected Area of Size S Around Destination with Intrusion Detection and Drops



Increased Drop Rate Near the Destination: Phase Transition Effect for Protection



What if the Energy Infrastructure were Designed like the Internet?

- Energy: *the* limited resource of the 21st Century
- Needed: Information Age approach to the Machine Age infrastructure
- Lower cost, more incremental deployment, suitable for developing economies
- Enhanced reliability and resilience to wide-area outages, such as after natural disasters
- *Packetized Energy?: Discrete units of energy locally generated, stored, and forwarded to where it is needed; enabling a market for energy exchange*

Imperial College London

New Energy Systems

- **A scalable energy *network* ?**
  - **Address inefficiencies at all levels of electrical energy distribution**
  - **Address energy generation and storage**
  - **IPS and PowerComm Interface**
  - **Energy sharing marketplace at small, medium, large scale**
- **Energy Supply on Demand**
- **Imagine some Test-beds: Smart buildings, datacenters**

Imperial College London

Some Publications

- O. H. Abdelrahman and E. Gelenbe. Time and energy in team-based search. *Phys. Rev. E*, 87(3):032125, Mar 2013.
- E. Gelenbe. Search in unknown random environments. *Phys. Rev. E*, 82(6):061112, Dec. 2010.
- E. Gelenbe. Energy packet networks: adaptive energy management for the cloud. *Proc. 2nd Inter'l Workshop on Cloud Computing Platforms (CloudCP'12)*, p. 1–5, Bern, 10 April 2012. ACM.
- Erol Gelenbe. Energy packet networks: smart electricity storage to meet surges in demand. In *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques (SIMUTOOLS'12)*, p. 1–7, Desenzano del Garda, 19-23 March 2012. ICST.
- E. Gelenbe and C. Morfopoulou. Power savings in packet networks via optimised routing. *ACM/Spirnger MONETS*, 17(1):152–159, 2012.
- E. Gelenbe and C. Morfopoulou. Gradient optimisation for network power consumption. In *First ICST International Conference on Green Communications and Networking (GreenNets 2011)*, 5-7 Oct 2011.

Publications

- E. Gelenbe and T. Mahmoodi. Distributed Energy-Aware Routing of Traffic. *Proc. 26th International Symposium on Computer and Information Sciences (ISCIS'11)*, London, UK, 26-27 Sep. 2011.
- E. Gelenbe and C. Morfopoulou. A Framework for Energy Aware Routing in Packet Networks. *The Computer Journal*, 54(6), June 2011.
- E. Gelenbe and T. Mahmoodi. Energy-Aware Routing in the Cognitive Packet Network. *International Conf. on Smart Grids, Green Communications, and IT Energy-aware Technologies (Energy 2011)*, Paper No. Energy_2011_1_20_50090, Venice, I22-27 May 2011.
- E. Gelenbe and C. Morfopoulou. Routing and G-Networks to Optimise Energy and Quality of Service in Packet Networks. *Proc. 1st Inter'l ICST Conference on E-Energy*, Athens, 14-15 Oct 2010.
- A. Berl, E. Gelenbe, M. di Girolamo, G. Giuliani, H. de Meer, M.-Q. Dang, and K. Pentikousis. Energy-Efficient Cloud Computing. *Computer Journal*, 53(7), 2010.
- E. Gelenbe and S. Silvestri. Reducing Power Consumption in Wired Networks. In *Proc. 24th International Symposium on Computer and Information Sciences (ISCIS'09)*, 14-16 Sep 2009, IEEE.

Imperial College London

Thank You !

http://san.ee.ic.ac.uk

Imperial College London