Rafael Tolosana
José Ángel Bañares
Engineering & Architecture School - Zaragoza University, Spain
banares@unizar.es, rafaelt@unizar.es

Omer F. Rana
School of Computer Science & Informatics - Cardiff University, UK
o.f.rana@cs.cardiff.ac.uk

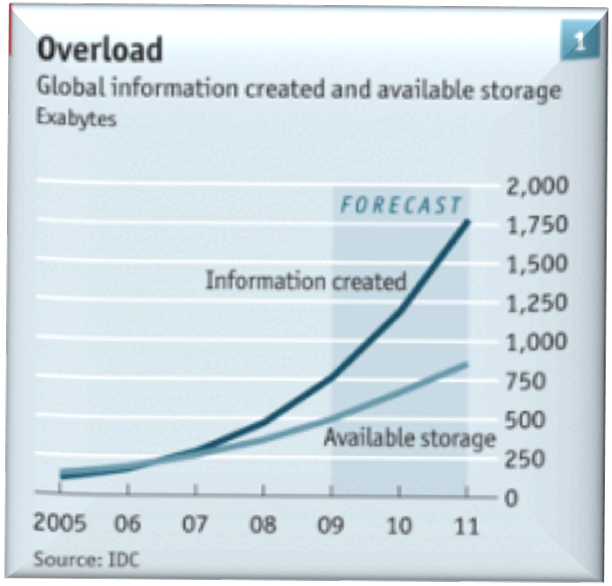
Congduc Pham
Laboratoire informatique- Université de Pau, France
congduc.pham@univ-pau.fr

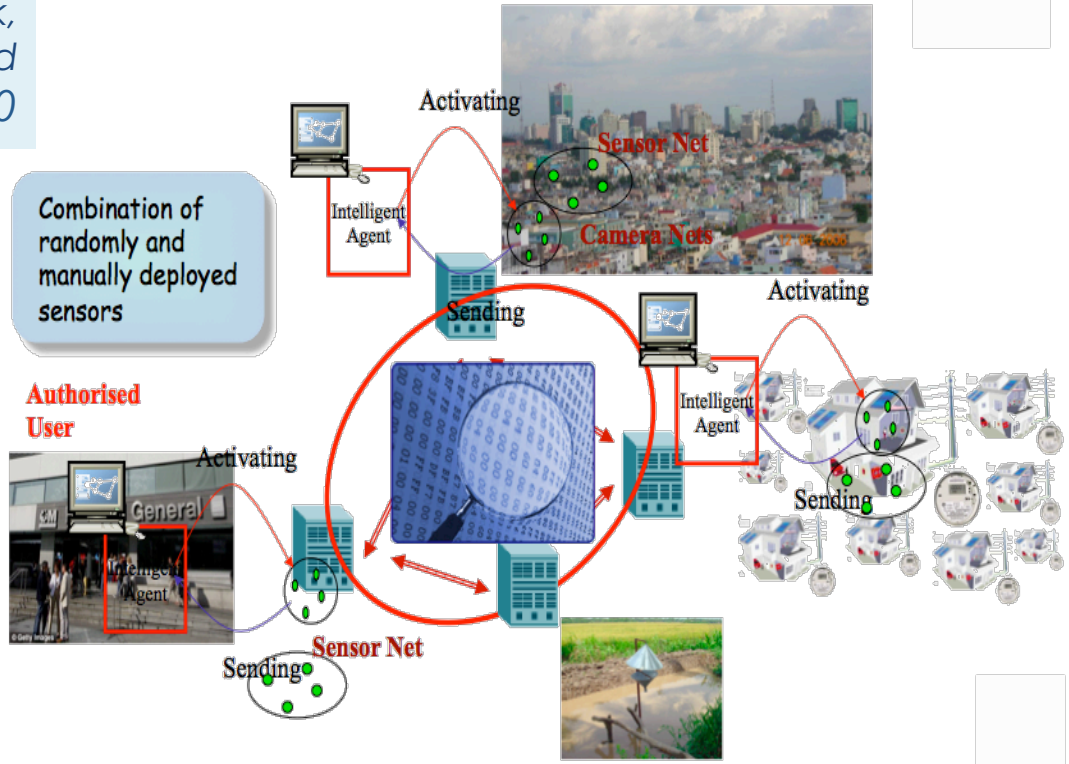
Revenue-based Resource Management on Shared Clouds for Heterogenous Bursty Data Streams

Towards global sensing

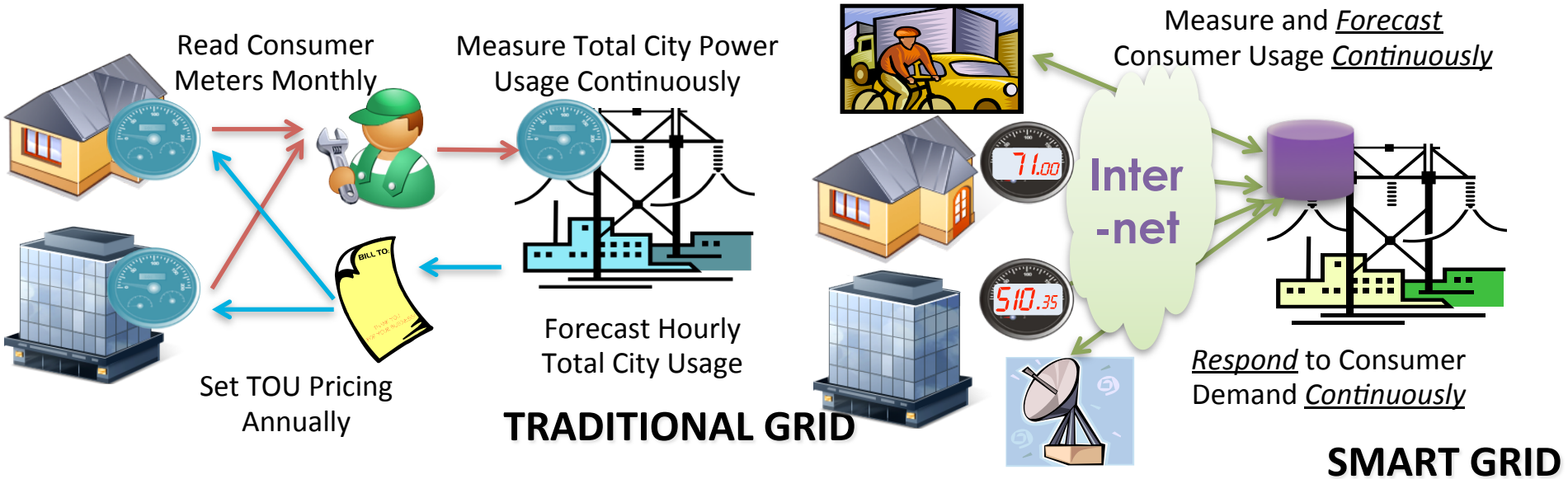
- Sensor networks, satellite surveys, high throughput laboratory instruments, observation devices, supercomputers, smart cities ...

Data Deluge *"Everywhere you look, the quantity of information in the world is soaring. The Economist, Feb 2010*





Ex. Transition to a *Smart* Grid



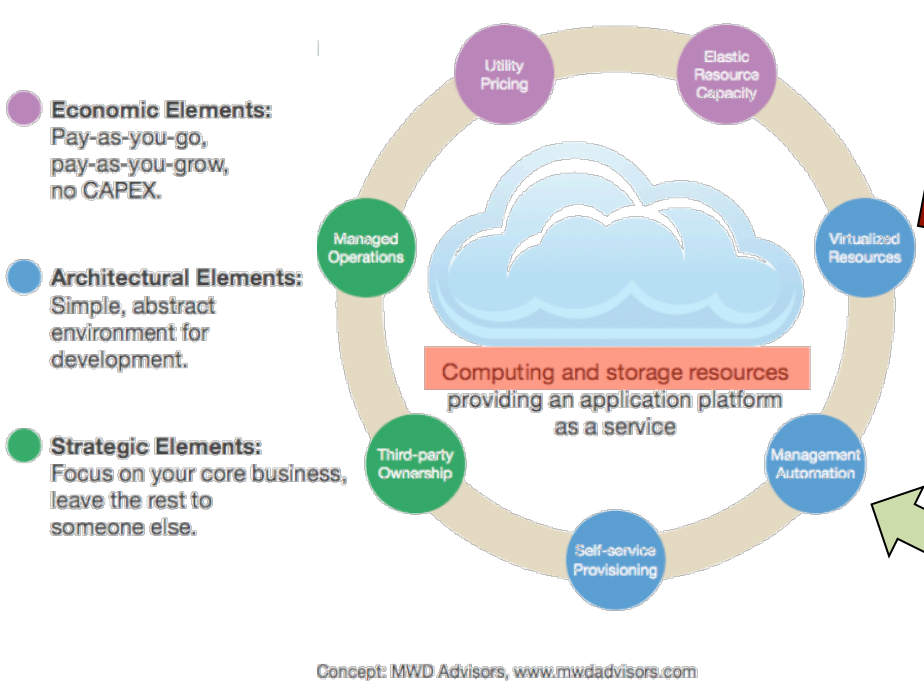
- **Smart meters**
  - Bi-directional, realtime communication between utility & consumer

Yogesh Simmhan (Centre for Energy Informatics, Univ. of Southern California)

The cloud assumption

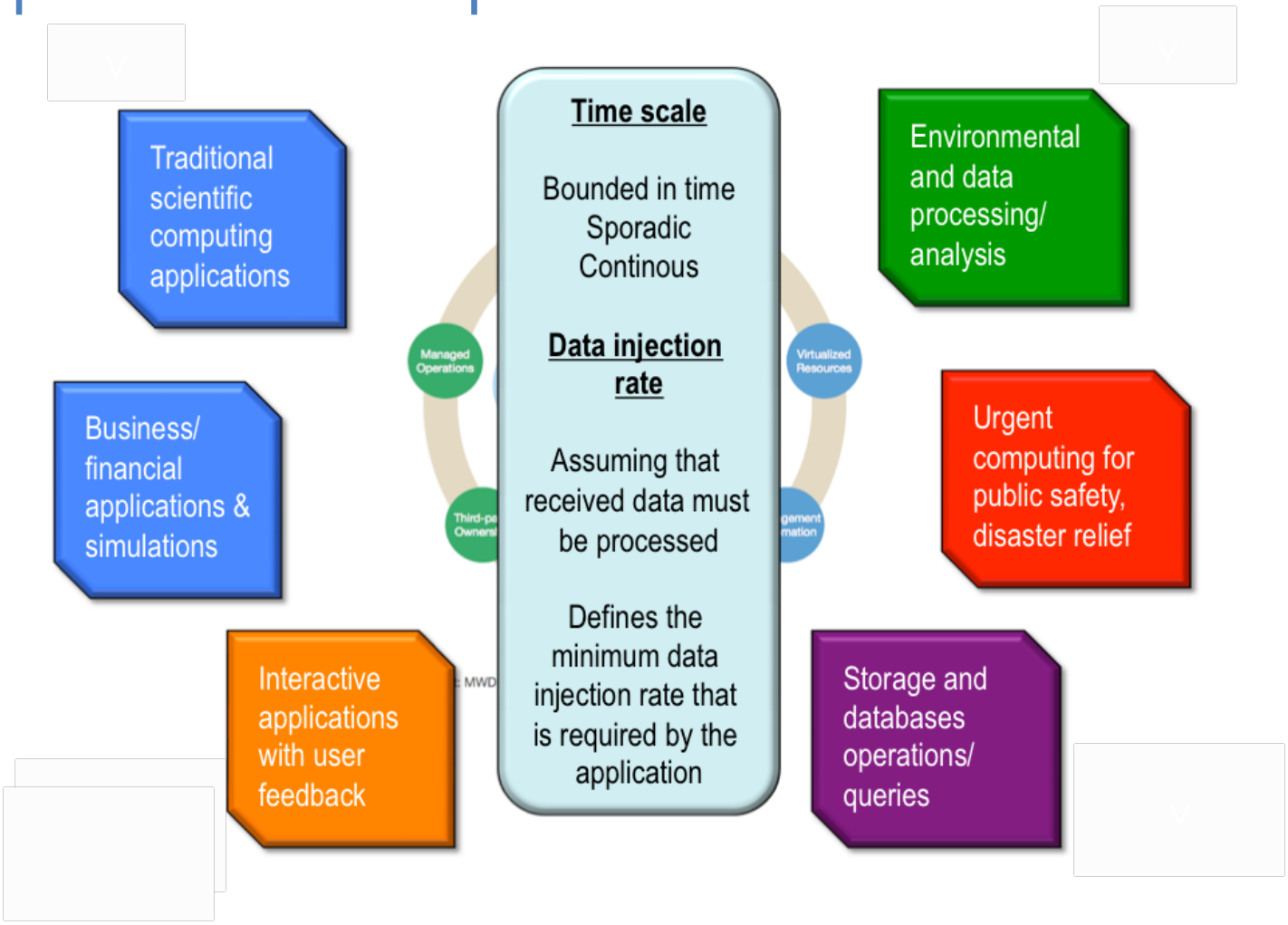
+ Different needs, same expectations

- **Preserving QoS applications on Shared Distributed infrastructures**
- **Expressing scalable solutions on heterogeneous infrastructures.**
- **Processing huge volume of data online**
- **Parallel processing of data**
- **Scaling data storage, network and computing resources**



Many users, with various profile, and different needs!

Application's profile



Approach & focus

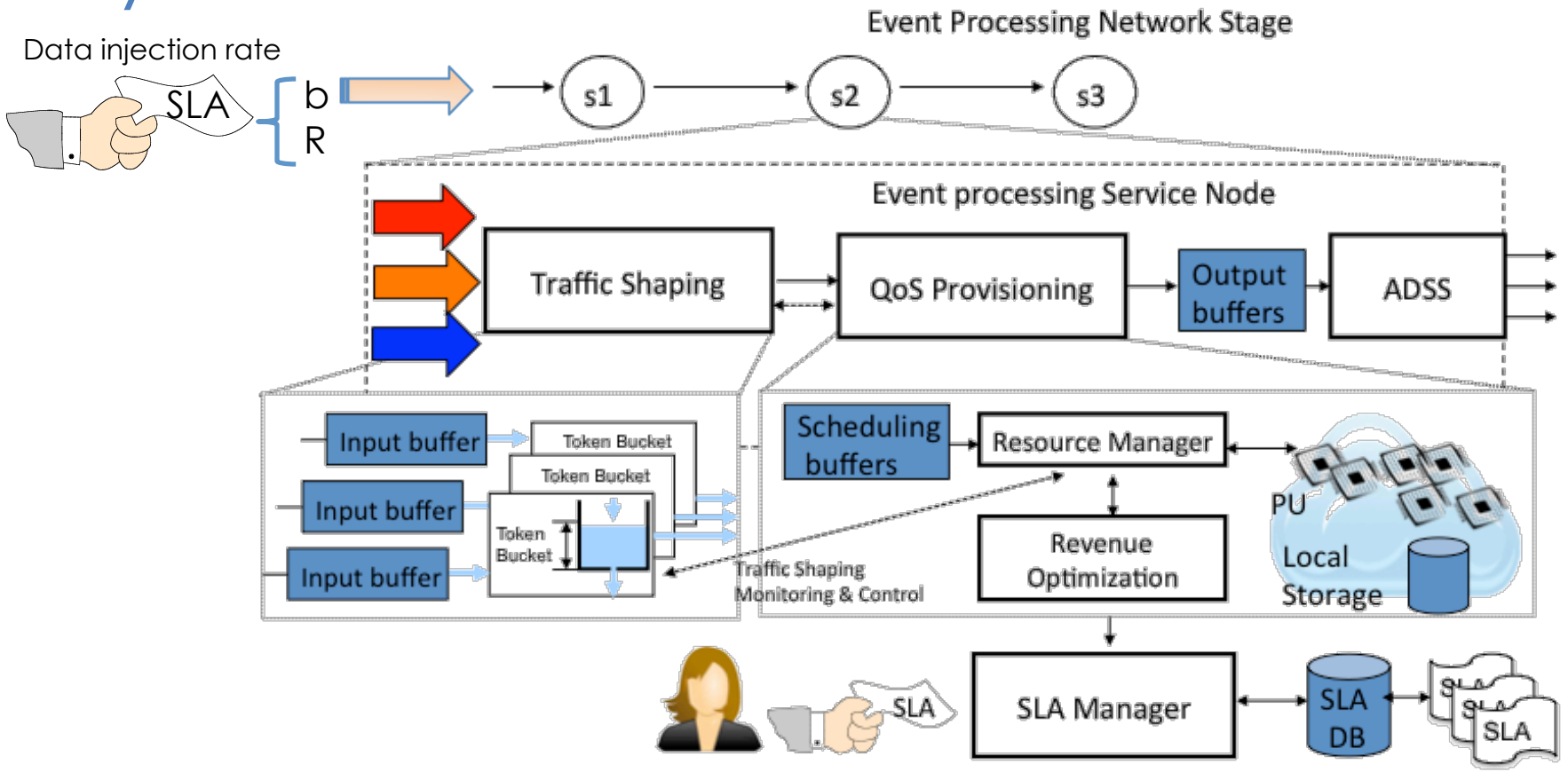
Adaptive infrastructure for sensor data analysis

- **Multiple** concurrent data streams with SLA
- **Variable** properties: rate and data types; various processing models
- Support for **in-transit** analysis, enforcing QoS
- Support for **admission** control & flow **isolation** at each node
- In case of QoS violation, **penalisation**

Key focus

- **Architectural components**
- **Business rules** for **SLA** Management : **Actions** to guarantee **QoS** & **maximize revenue**

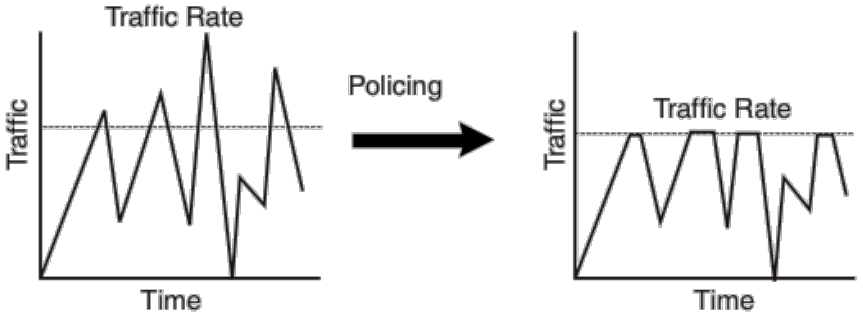
System Architecture



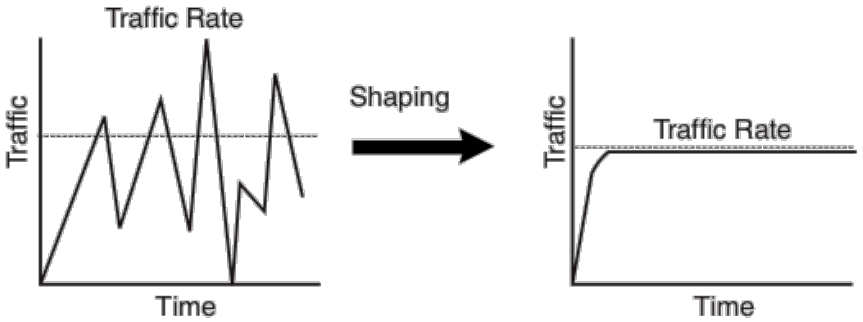
- 3 key components / node: Token Bucket, Processing Unit & output streaming

Token Bucket (shaping traffic)

Traffic shaping component allows **to control the traffic** going out this component in order **to match** its **flow** to the processing speed of **available resources** and to ensure that the traffic conforms to policies contracted for it

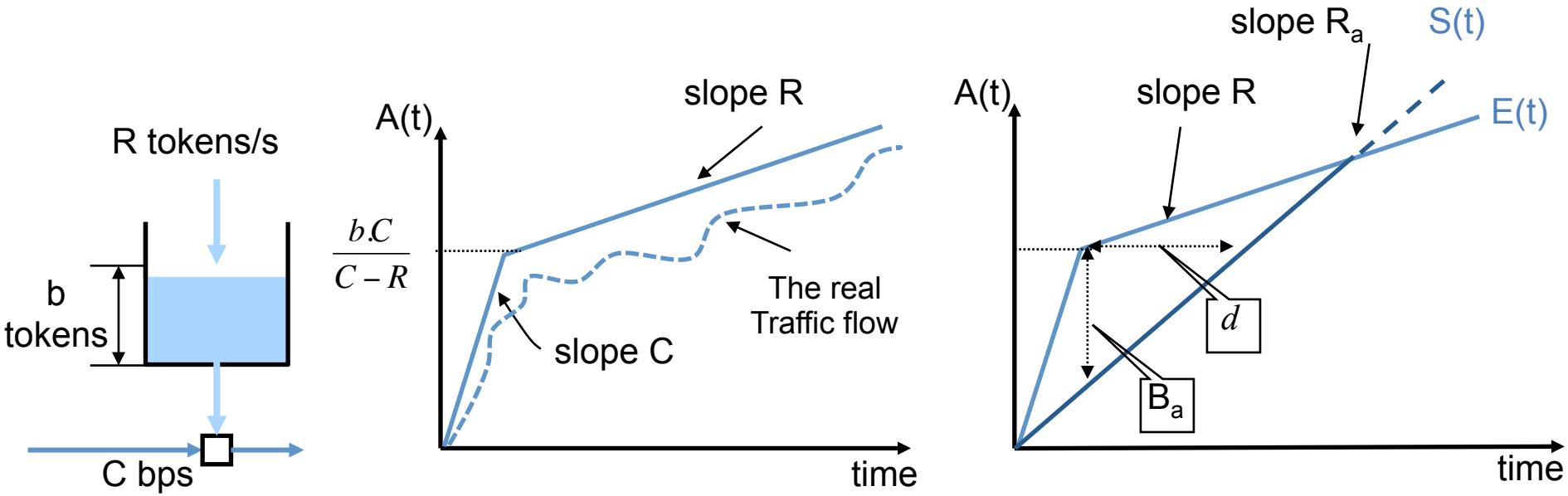


- A **policer** typically drops excess traffic.



- A **shaper** typically delays excess traffic using a buffer to hold data and shape the flow when the data rate of the source is higher than expected.

Token Bucket (shaping traffic)

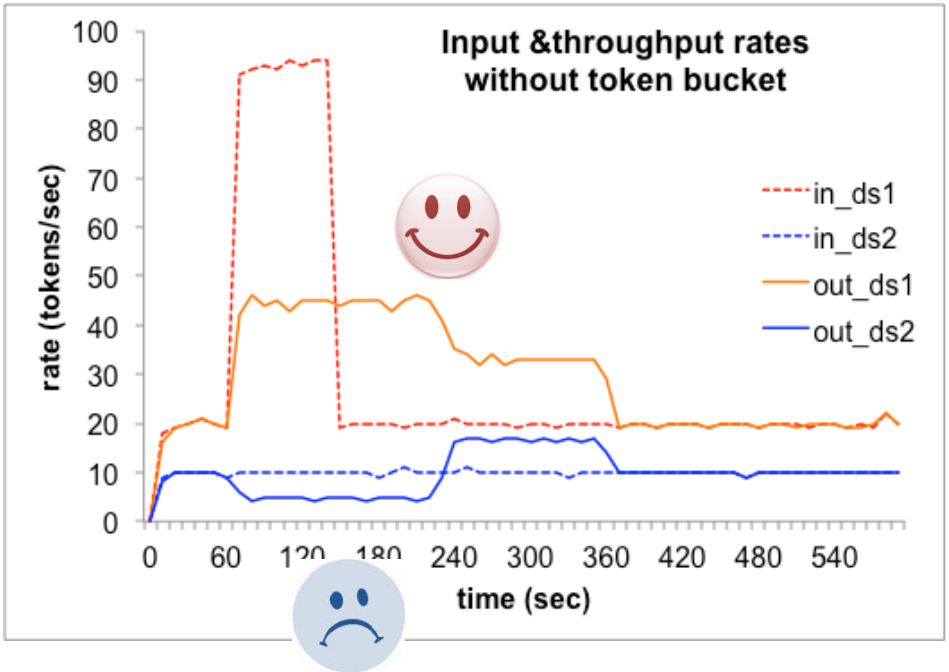


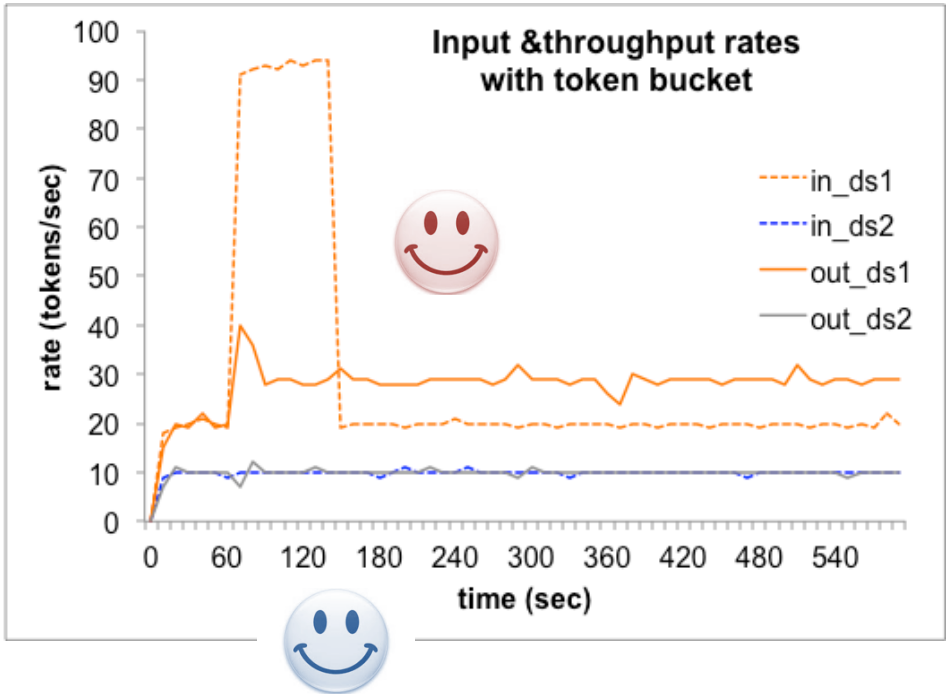
A(t): Amount of data arriving up to time t

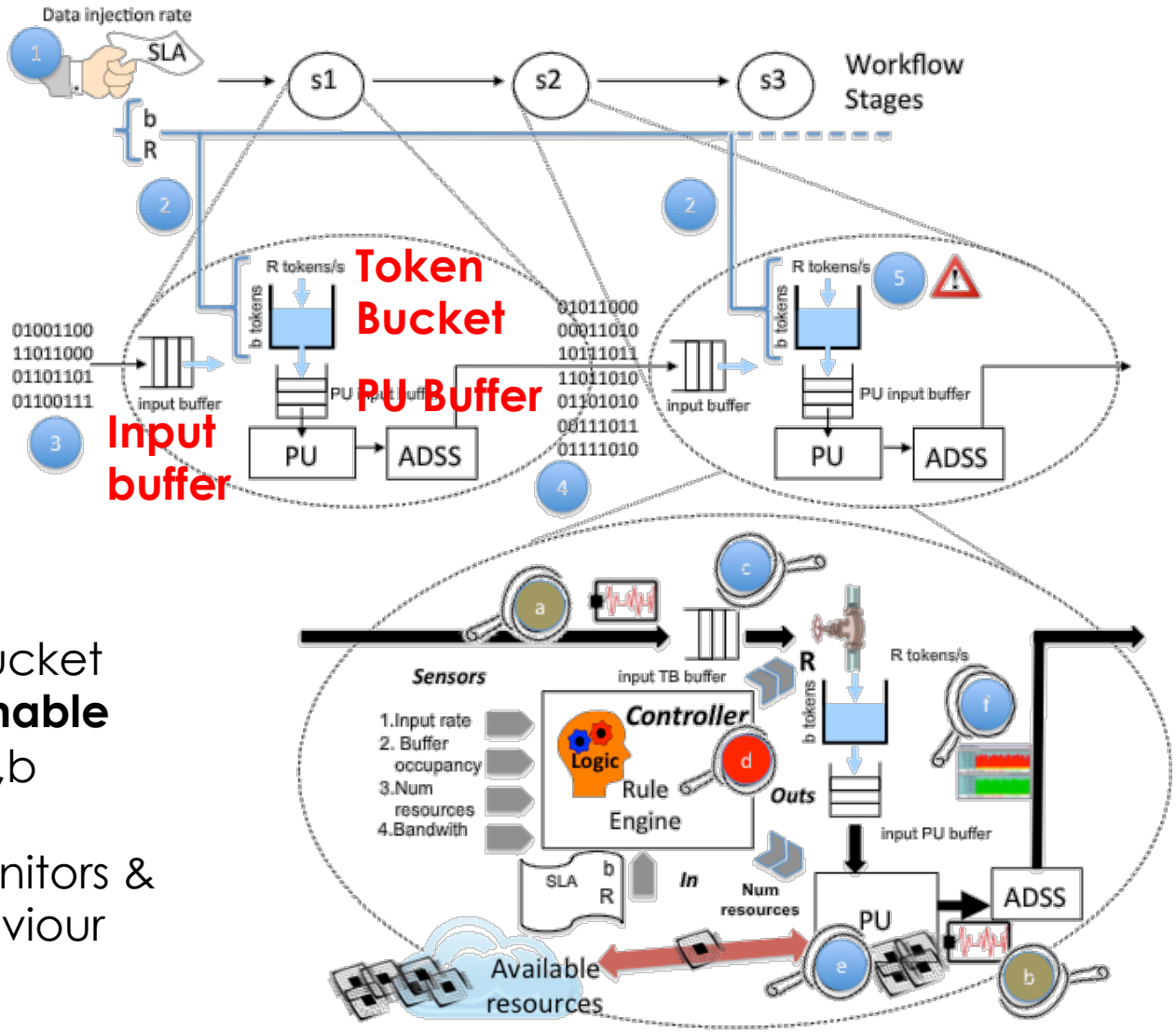
Two key parameters of interest:

- **R**: Also called the **committed information rate** (CIR), it specifies how much data can be sent or forwarded per unit time on average
- **B**: it specifies **for each burst how much data can be sent** within a given time without creating scheduling concerns

Token Bucket (shaping traffic)







Each token bucket provides us **tunable** parameters: R,b

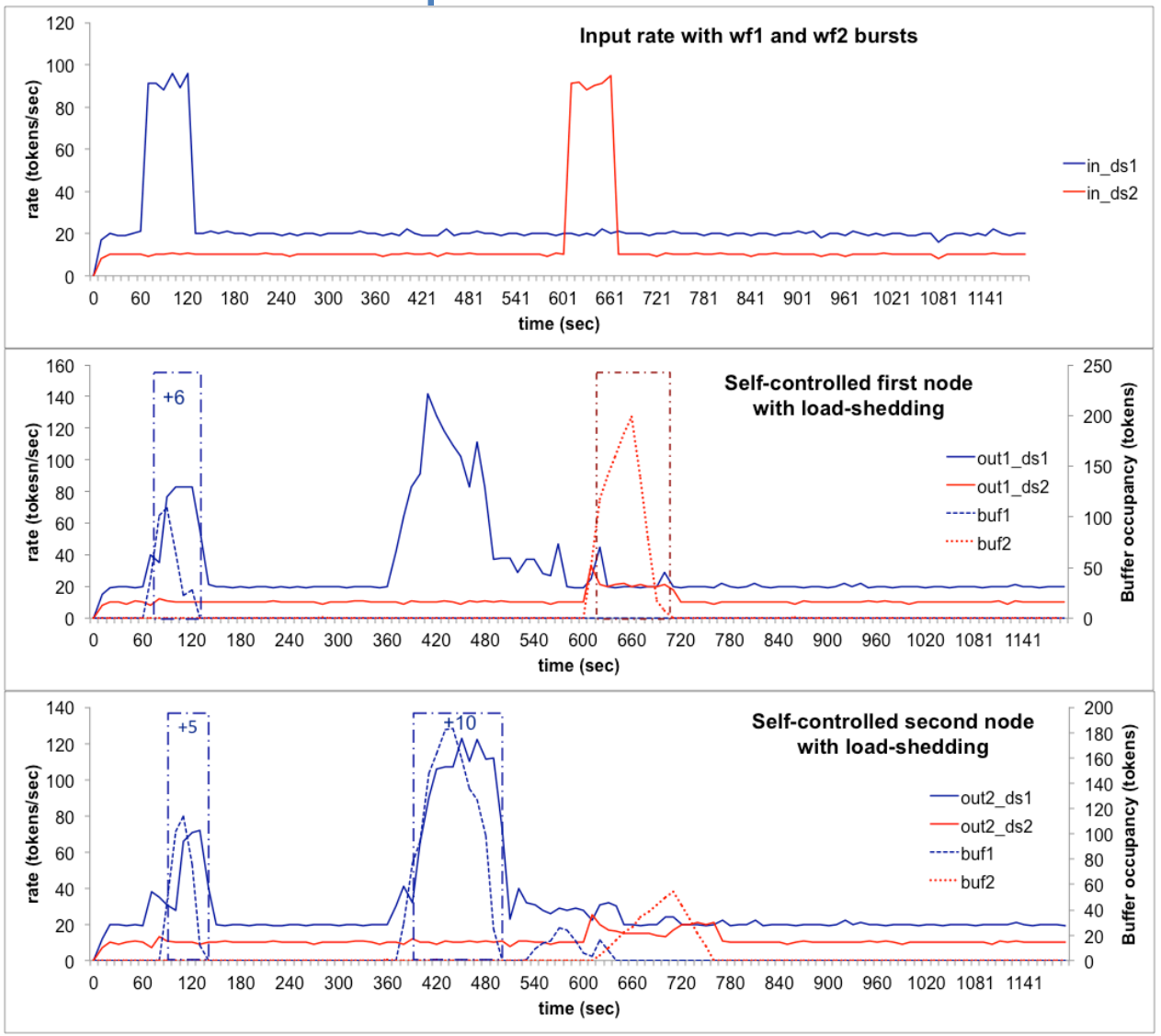
Controller: monitors & modifies behaviour

Control for Elastic SLA definitions

Controller: monitors & modifies behaviour

- **Token bucket behaviour** is regulated by **b, R** parameters
- **SLAs** can specify **more flexible behaviours** allowing the **controller** to take different **actions** when a threshold is reached
  - **Load-shedding:** drop data stored by the token bucket buffer
  - **Modify** the mean injection rate **R**

Flexible SLAs with drop and R control



Control for the Revenue Model

Elements of the model

- **Revenue:** price charge to n clients for m operations

$$\sum_{i=1}^{n} \sum_{j=1}^{m} Pr(O_{ij})$$

- **Cost:** for performing such operations $c(O_{ij})$
- **Penalisation:** in case of QoS violation for client $PSLA_{ij}$

Maximizing provider revenue

$$\sum_{i=1}^{n} \sum_{j=1}^{m} Pr(O_{ij}) - \sum_{i=1}^{n} \sum_{j=1}^{m} min(c(O_{ij}), PSLA_{ij})$$

Revenue Model

SLA for each data stream

i. A desired **QoS level** for each operation $L_{desired_{ij}}$

ii. The minimum QoS level acceptable $L_{min_{ij}} \leq L_{desired_{ij}}$

iii. The cost $c(O_{ij})[k]$ for each QoS level defined by the service in the range $[L_{min_{ij}}, L_{desired_{ij}}]$

iv. A penalty $PSLA_{ij}$ when it fails to meet the minimum level

Maximizing provider revenue

- Select an **optimal QoS level** for each operation
- The **aggregated number** of resources required to provide each operation Oij [k] does not exceed available resources

Revenue Model

- QoS requirements are often defined using the **worst case scenario**
- **Business policies** can be used to improve revenue and to provide flexibility in SLA definition
  - **Under-provisioning** of resources
  - selective **SLA violations**
  - Increase in the number of computational resources
    - **Computational resources can be borrowed** from less prioritized data flows
  - **load shedding** (when possible)

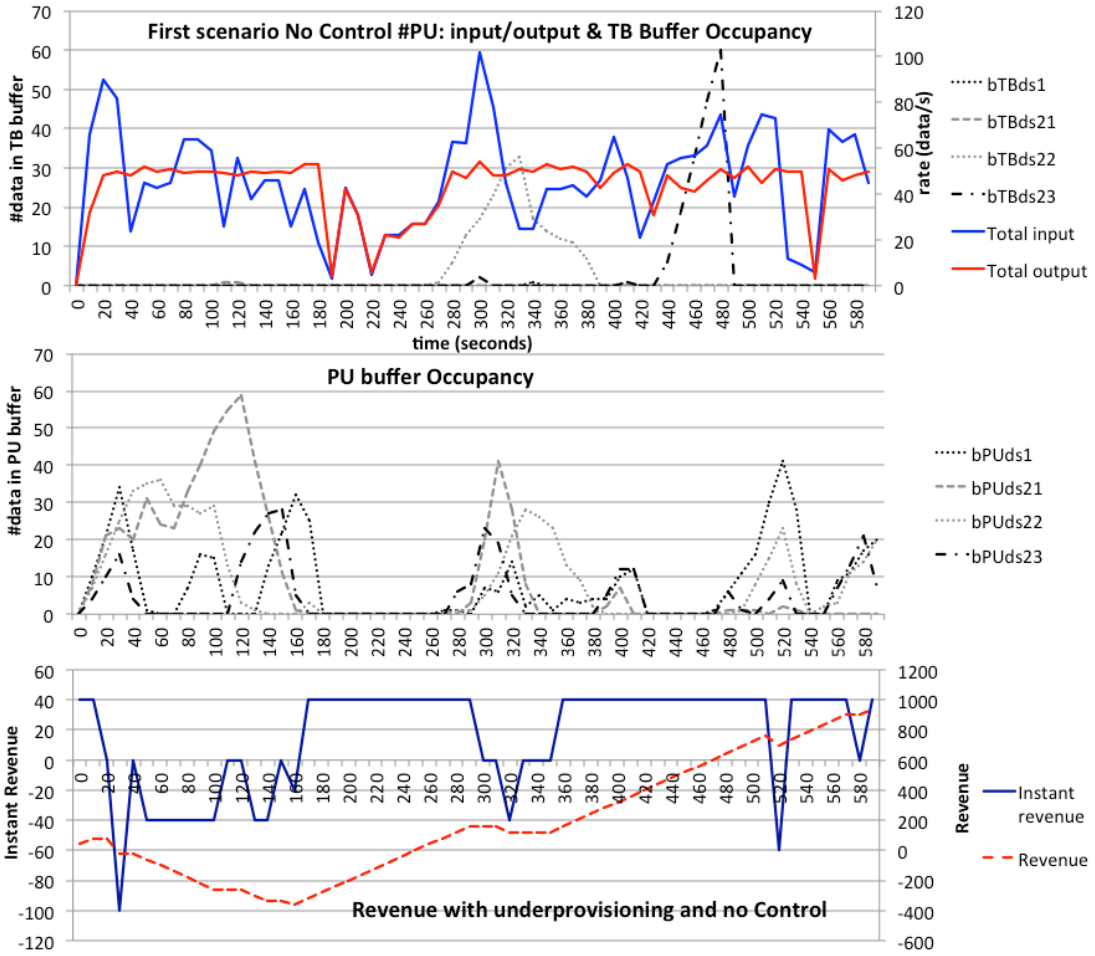
Classification of Clients

- **Gold** – for high penalty and revenue
- **Silver** – for medium penalty and revenue
- **Bronze** – for low revenue and no penalty

Business rules

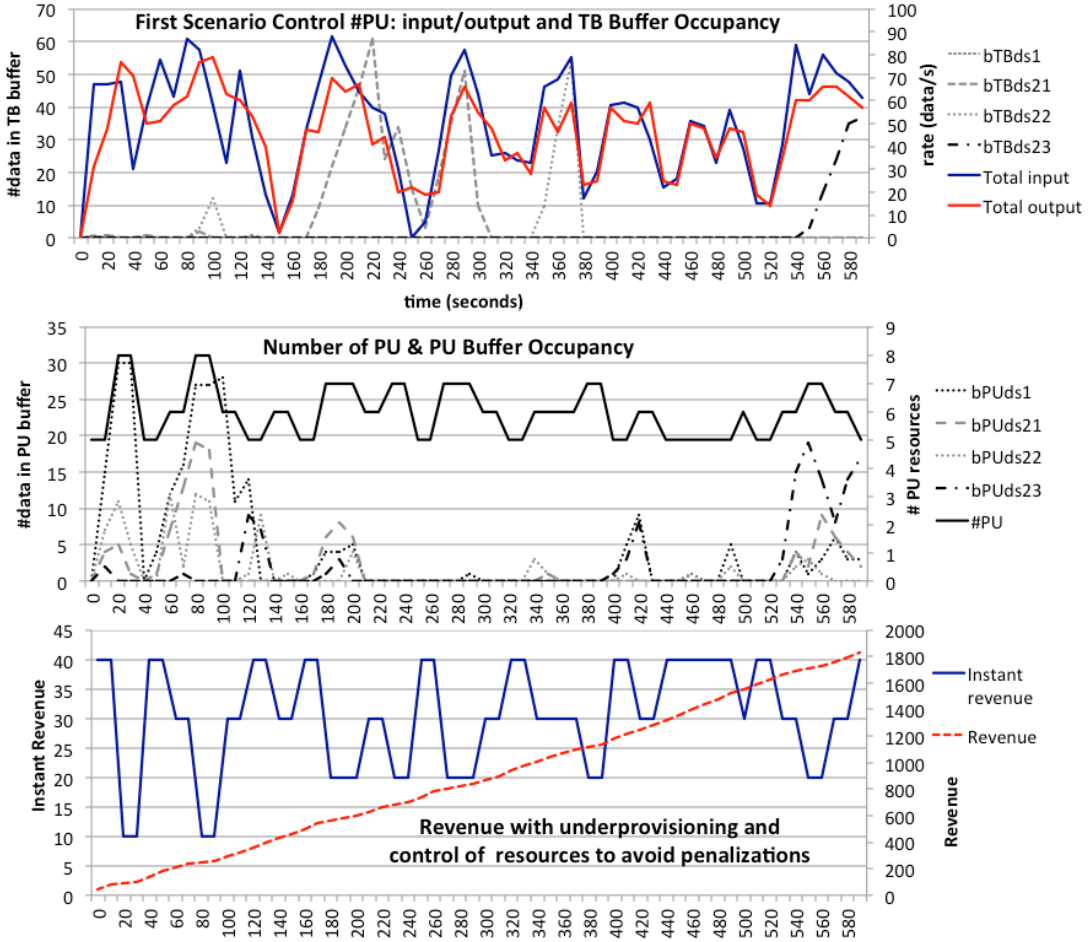
	Pattern	Action
	Data flow control	
1	**E:** B_i over threshold **C:** SLA_i allows control the use of free resources	$\Delta R_i = \sum_{i=1}^{n} NumRes_i * \hat{\delta_i} - \sum_{i=1}^{n} R_i$
2	**E:** B_i over threshold **C:** SLA_i allows control to drop D_i	$B_i = B_i - D_i$
3	**E:** B_i above threshold **C:** Controlled Stream	$\Delta R_i = 0$
	Ranges of QoS control	
4	**E:** $\sum_{i=1}^{n}(\lambda_i - R_i)$ over threshold **C:** QoS level allow to borrow N_i resources	$\Delta NumRes = min(\sum_{i=1}^{n} N_i, \sum_{i=1}^{n}(\lambda_i - R_i)/\hat{\delta_i})$
5	**E:** $\sum_{i=1}^{n}(\lambda_i - R_i)$ over threshold **C:** Queue level allow to pause low level data flows	$\#Paused_{LowLevel} = \sum_{i=1}^{n}(\lambda_i - R_i)/\hat{\delta_i}$
6	**E:** Overthrow **C:** Controlled Stream	$\Delta NumRes = 0, \#Paused_{LowLevel} = 0$

I Scenario (under-provision)
Golden takes additional resources



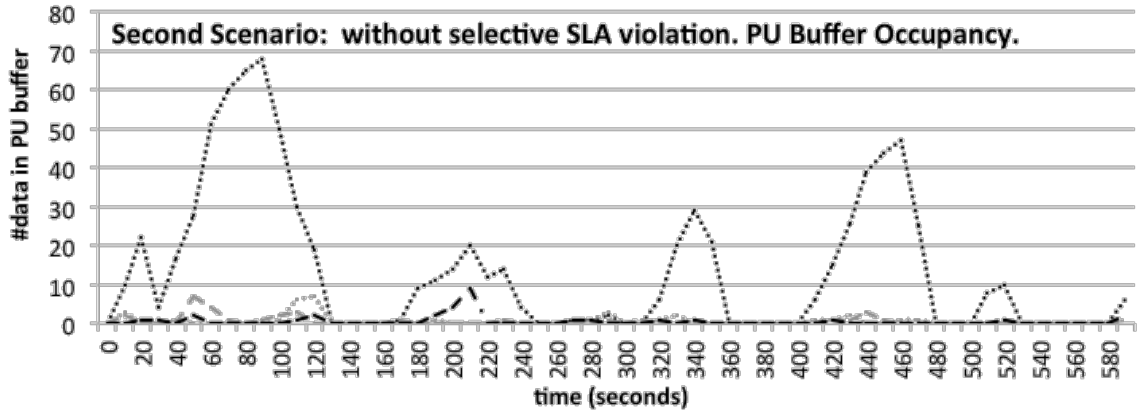
NO ADDITION OF RESOURCES FOR GOLDEN

I Scenario (under-provision)
Golden takes additional resources

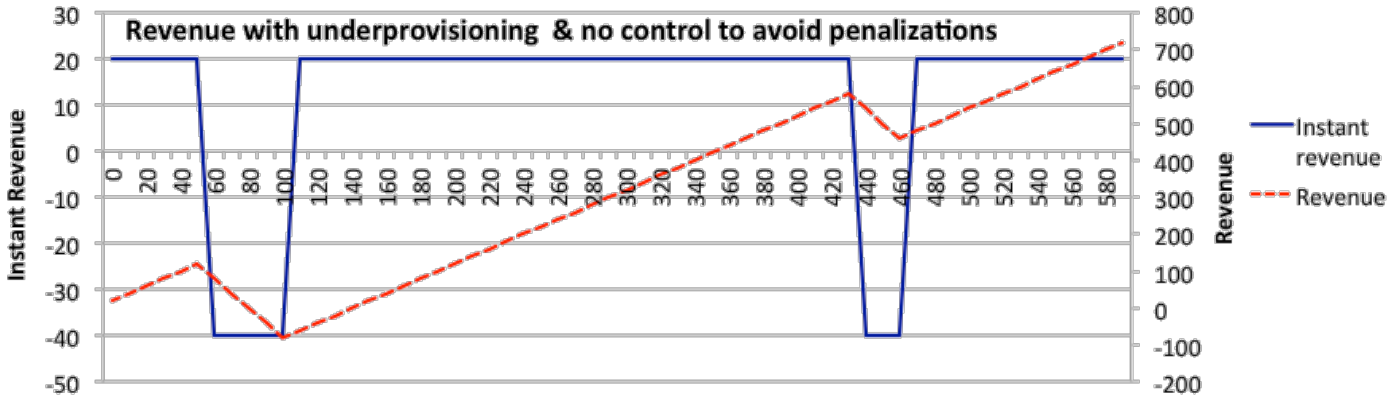


ADDITION OF
RESOURCES FOR GOLDEN

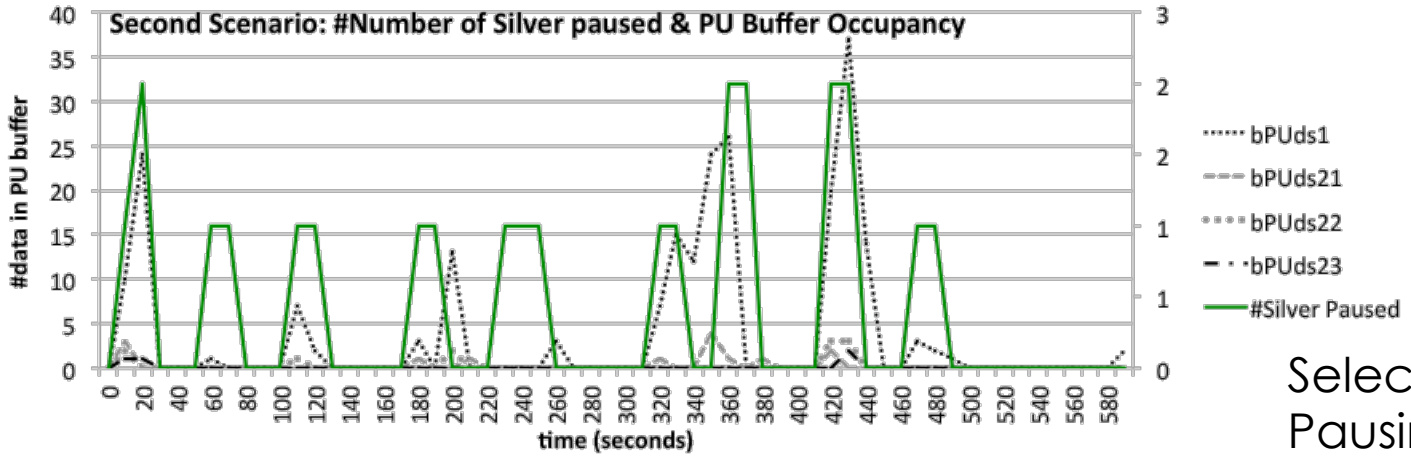
II Scenario
Selective Violation of SLA

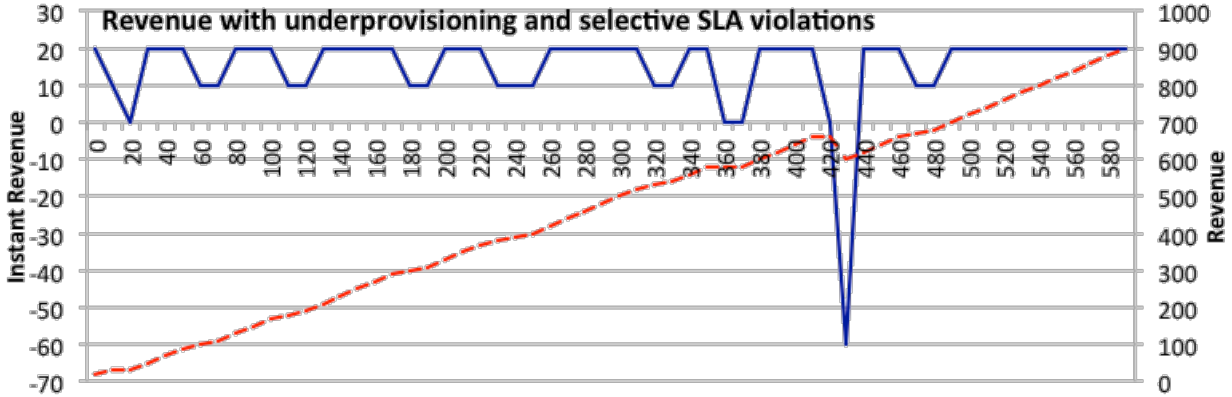


No Selective Violation



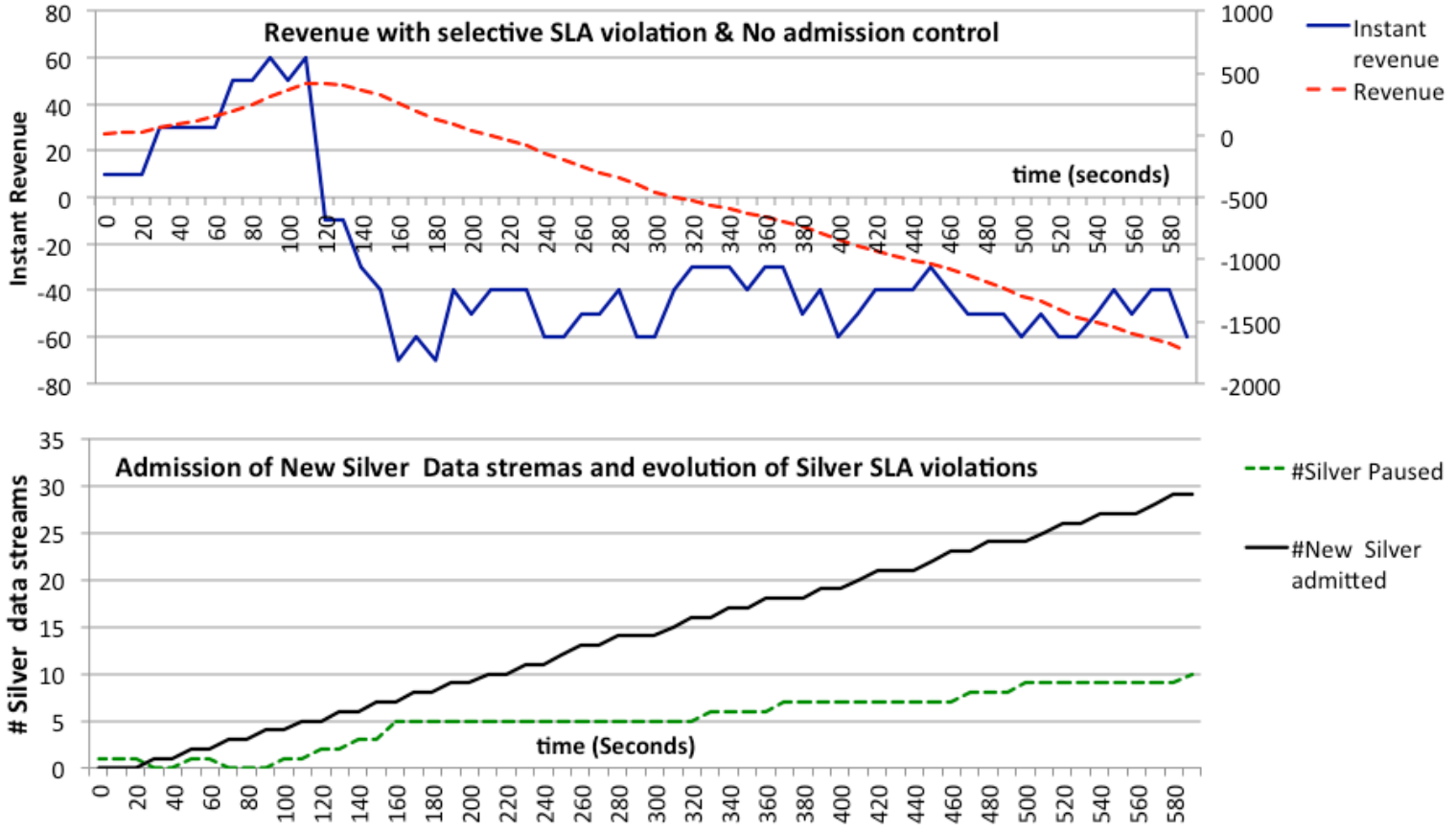
II Scenario
Selective Violation of SLA





Selective Violation
Pausing Silver clients

III Scenario
Need of Control Admission



Conclusion & Future work

Resource Management for Bursty Data Streams.

- **SLAs** can be specified in a flexible way by **Token bucket parameters**
- **Adapting Token bucket parameters** allow the definition of more flexible SLAs.
- **Business rules** can complement optimization process to improve revenue.

Future Work

- **Develop efficient mechanisms**
  - to implement previous Business strategies
  - To predict and avoid penalizations
- **Validation in real scenarios**: Smart grid, forecasting Electrical Vehicles charge
- **Classification** of clients in categories

Rafael Tolosana
José Ángel Bañares
Engineering & Architecture School - Zaragoza University, Spain
banares@unizar.es, rafaelt@unizar.es

Omer F. Rana
School of Computer Science & Informatics - Cardiff University, UK
o.f.rana@cs.cardiff.ac.uk

Congduc Pham
Laboratoire informatique- Université de Pau, France
congduc.pham@univ-pau.fr

Revenue-based Resource Management on Shared Clouds for Heterogenous Bursty Data Streams

GECON 2012