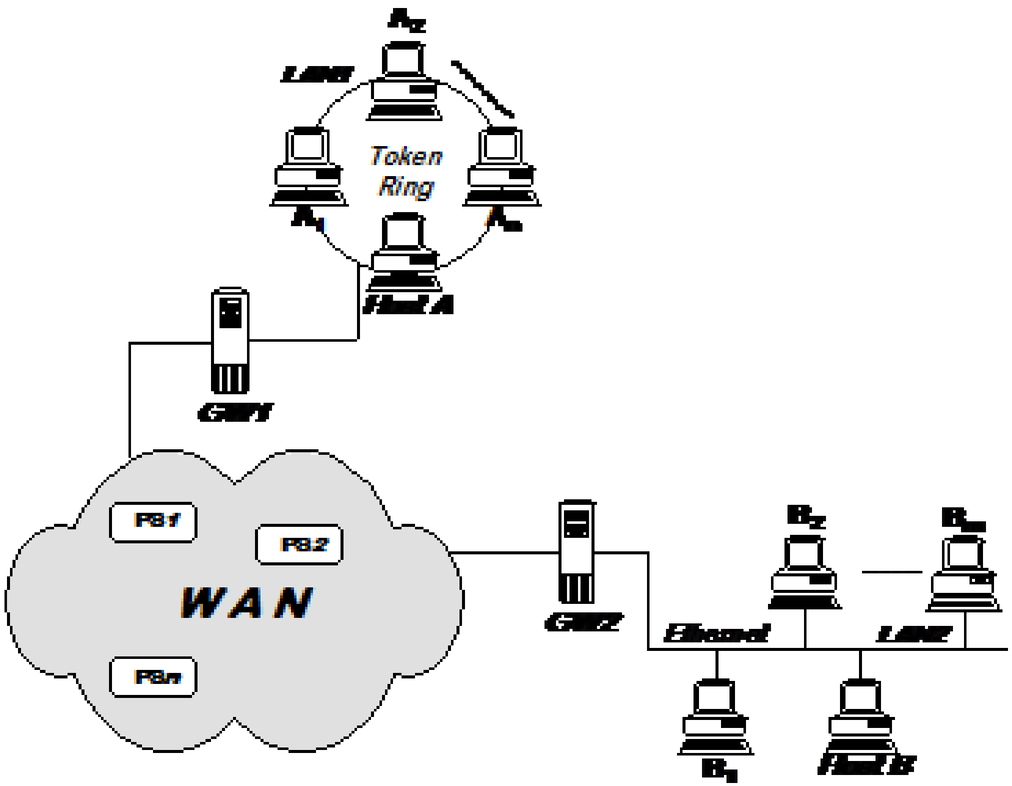
Vision of Internet platform and QoS Framework

Objective

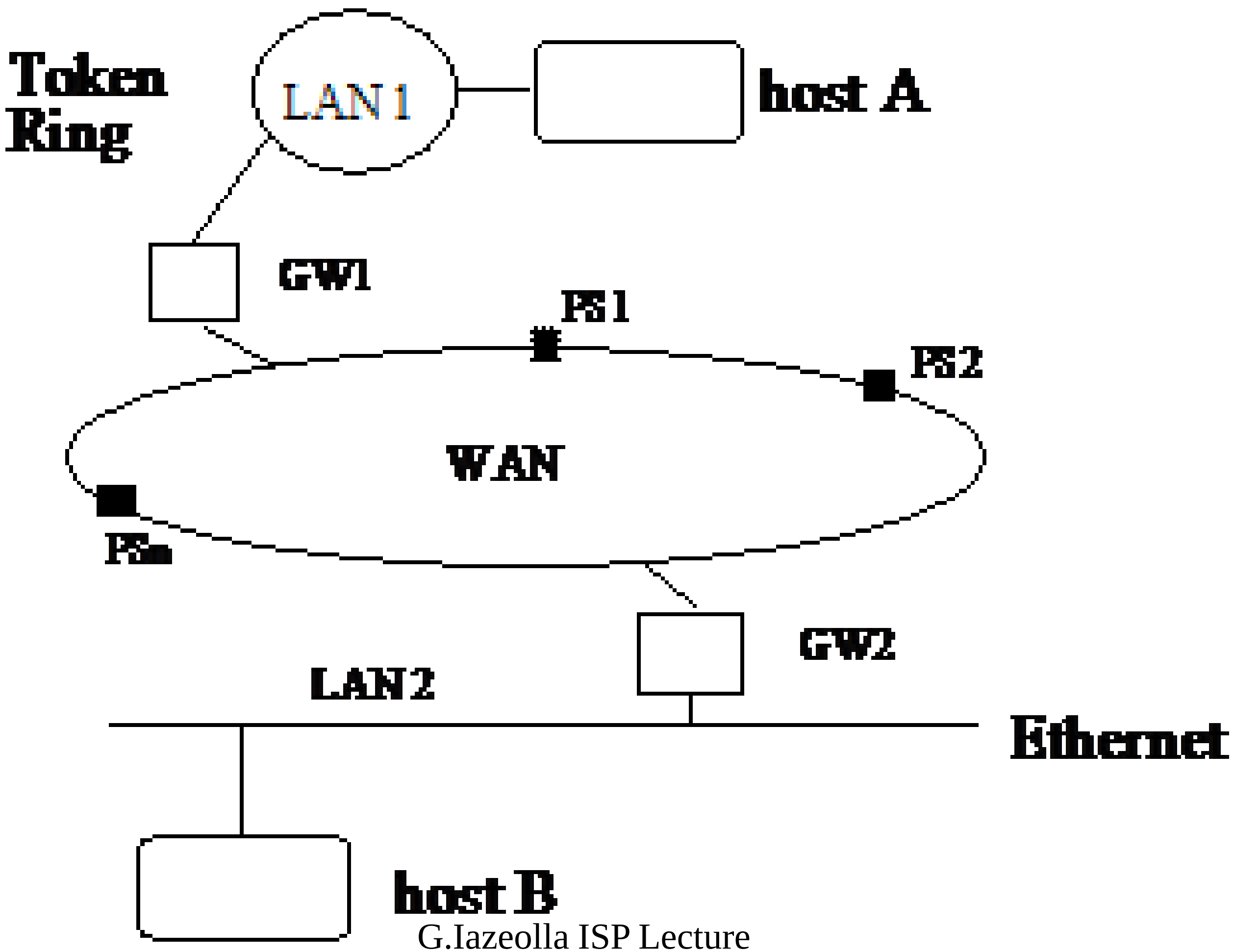
- To study performance and QoS of the platform
  - Response time
  - throughput
  - end-to-end delay

Multihost vision: internet platform
(PS= packet switch node in WAN)



Single-host vision: internet platform

(PS= packet switch nodes in WAN)



Single host vision and internet user platform

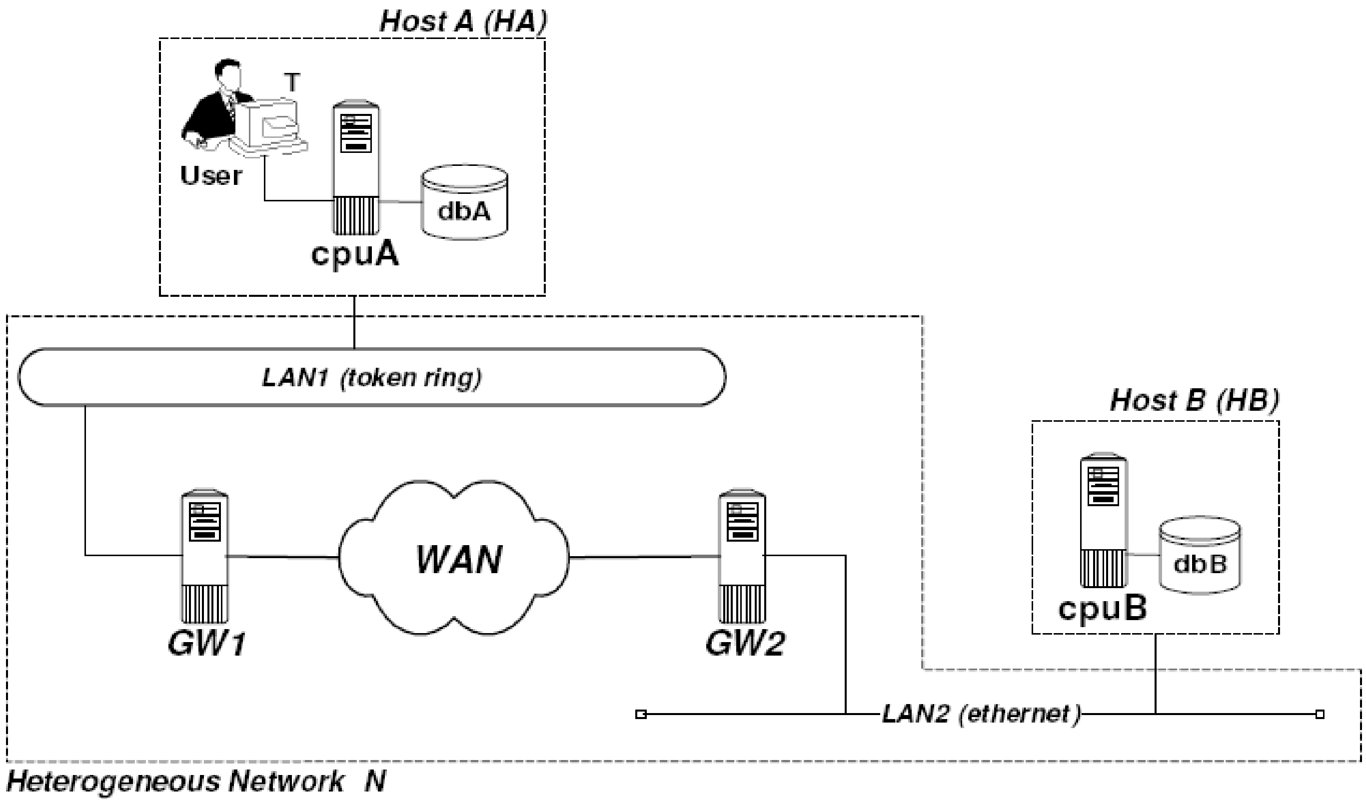
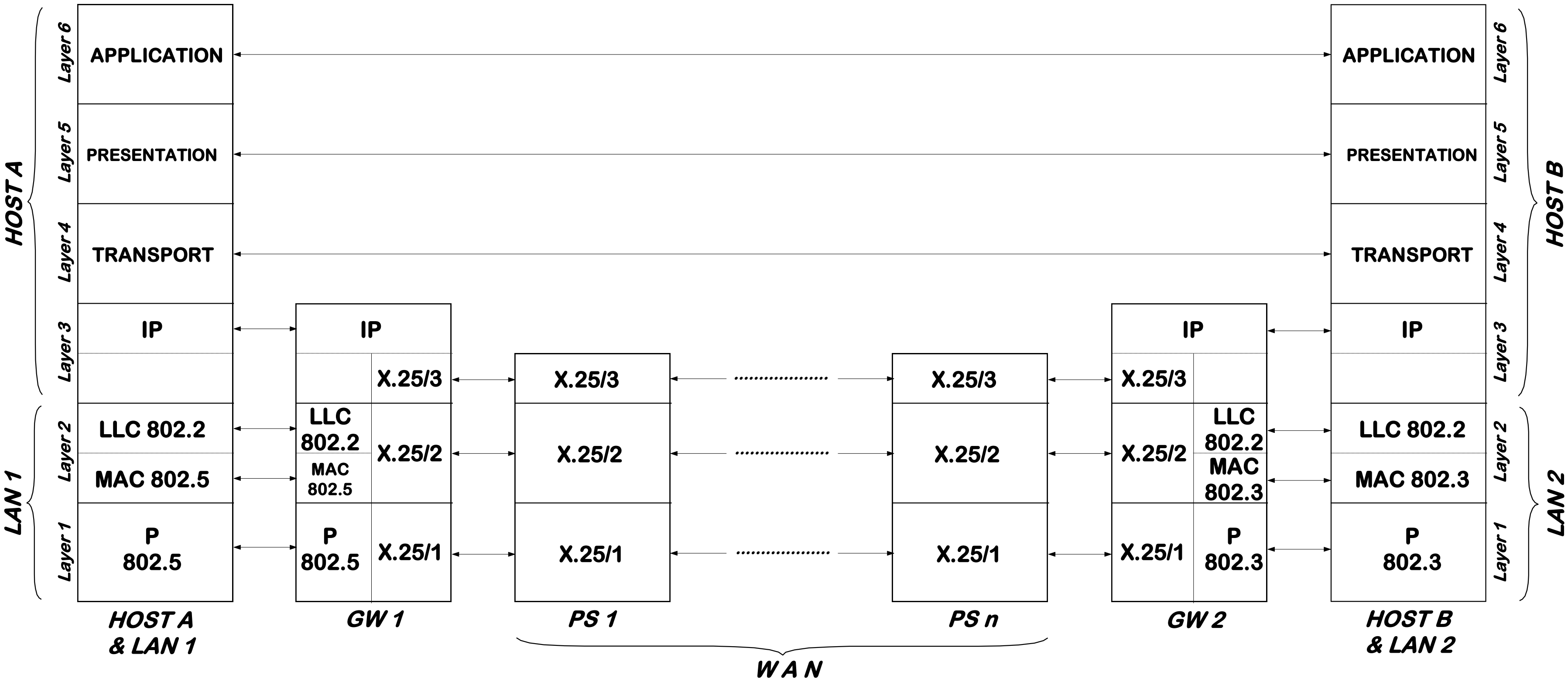


Fig. 1. General view of the system platform

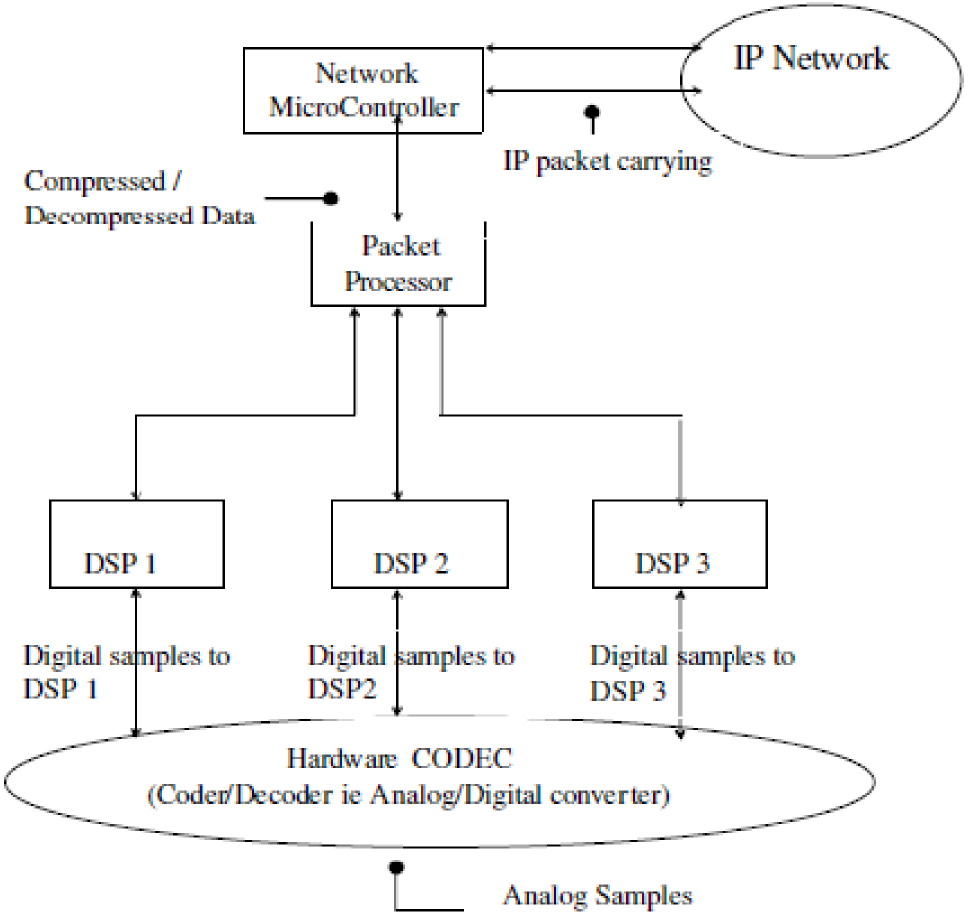
Protocol recall



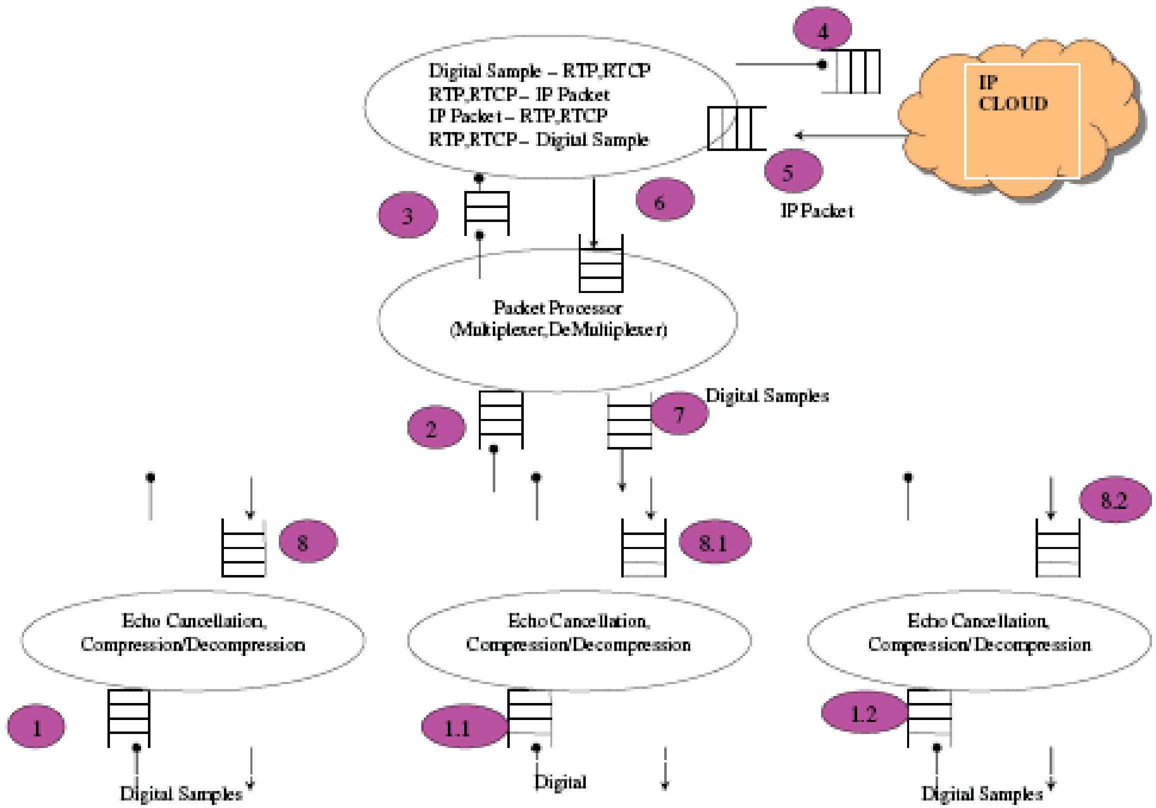
Description of protocols

- Layer 6: Application layer
- Layer 5: Presentation layer
- Layer 4: Transport layer (protoc. **TCP**)
- Layer 3: Network layer (protoc. **IP**)
- Layer 2: Distinto in
  - **L**ogical **L**ink **C**ontrol sub-layer (protoc. **LLC**) e
  - **M**edia **A**ccess **C**ontrol sub-layer (protoc. **MAC**)
- Layer 1: Physical layer (protoc. **P**)

Architecture of a VoIP Gateway

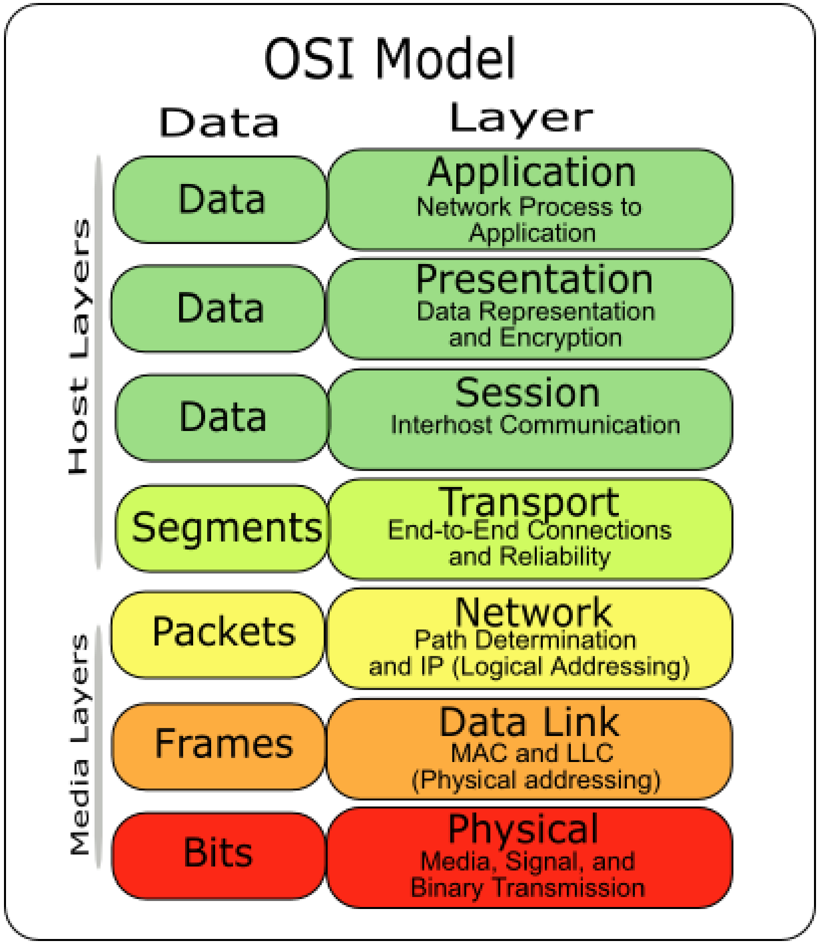


Performance model of a VoIP Gateway

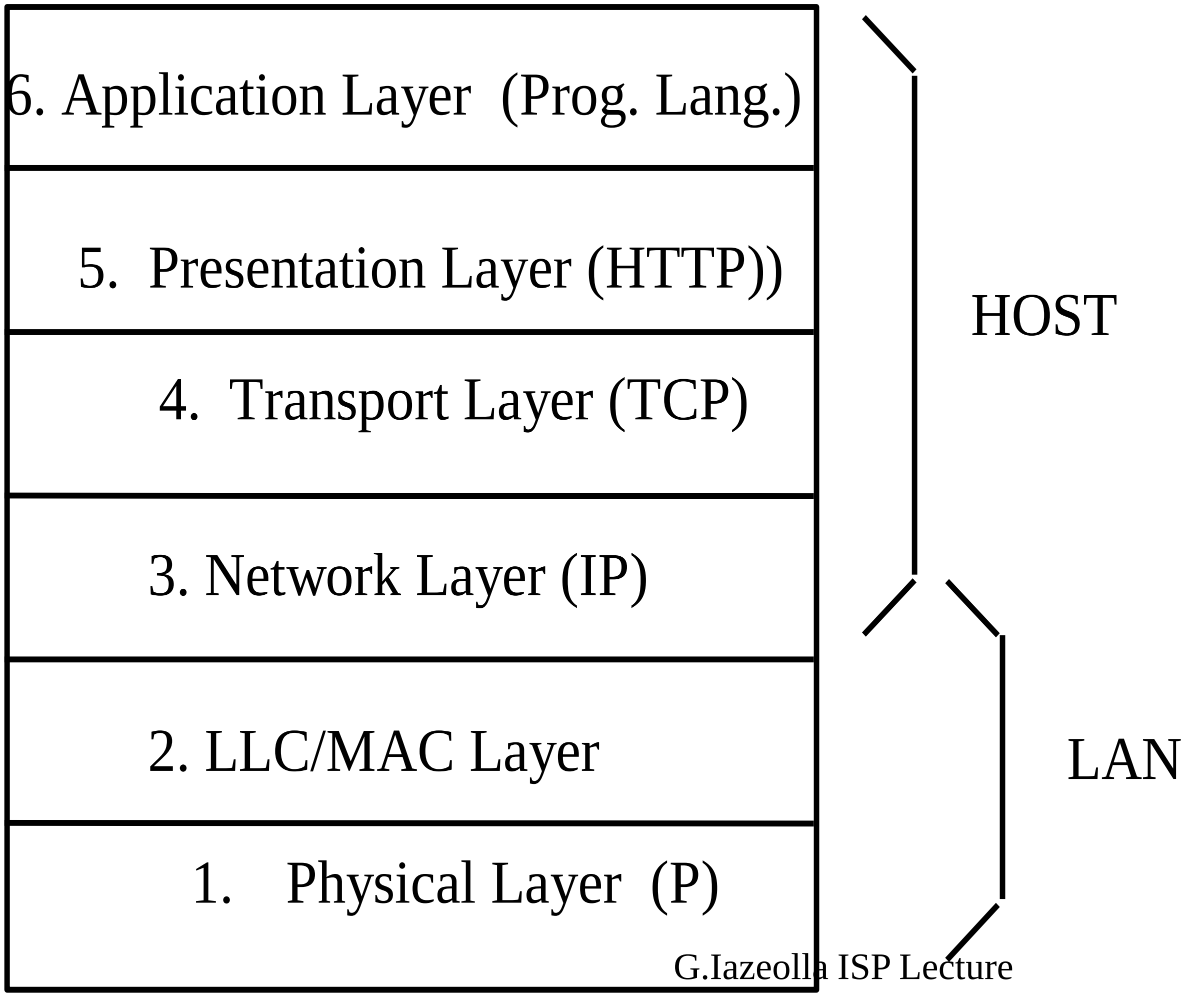


The OSI (Open System Interconnection) reference model (also known as the ISO/OSI model)

Vision of the Host Layers and the Media Layers

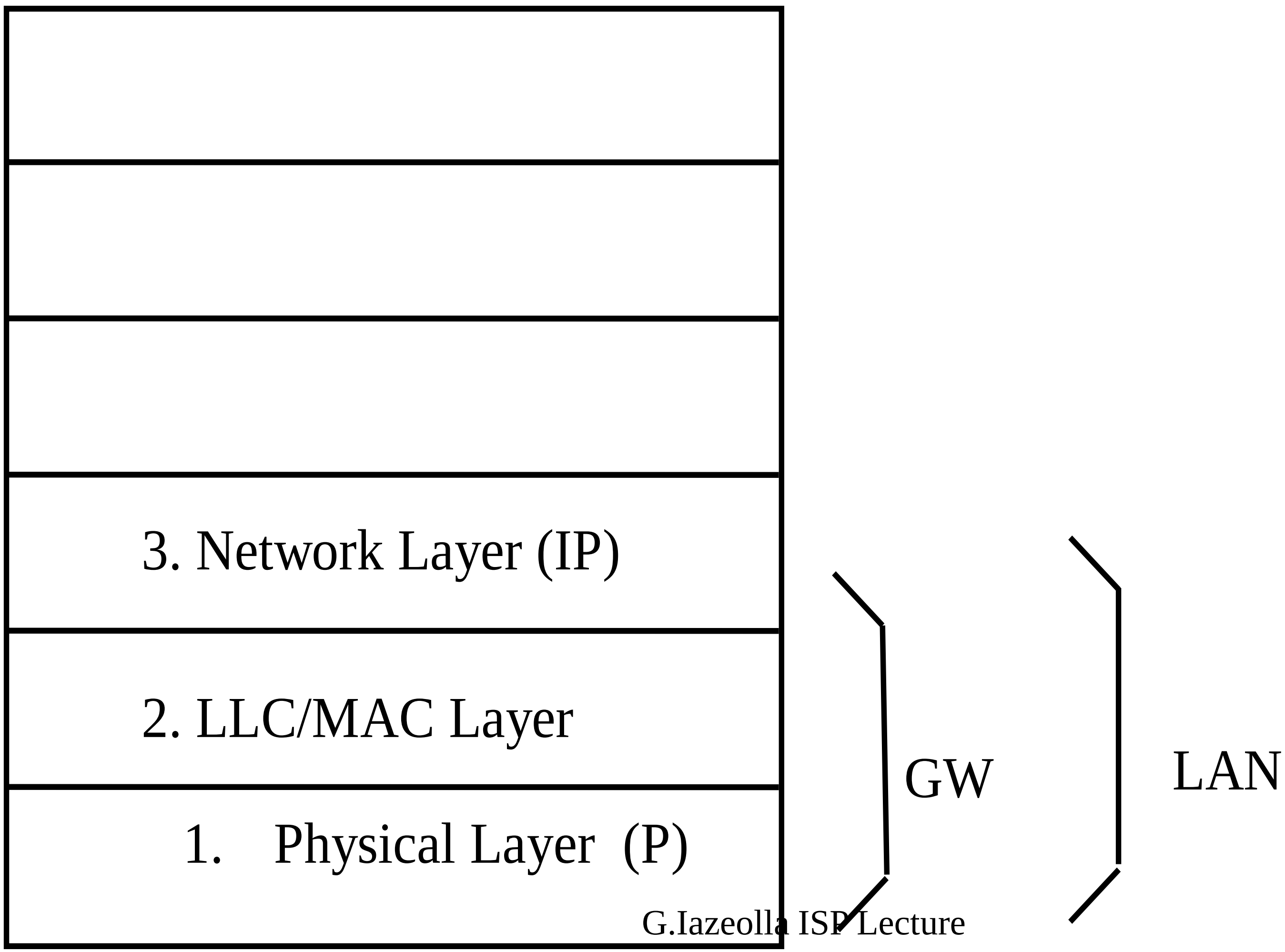


vertical description of the network functions
(from HOST towards LAN)



vertical description of the network functions

(from LAN towards Gateway)



Vertical description of the network functions

Through virtual machines *(or **abstract**)*

Functionality of the network described by the various abstraction levels

from the bottom to the top: progressive ***virtualization*** of the network

VM_1 hardware or physical machine (*circuit level*) *protoc. P*

VM_2 firmware machine (*microprogram language network level*) *protoc. MAC/LLC*

VM_3 network layer machine (*network language level*) *protoc.IP*

VM_4 transport layer machine (network *operating system language level*) *protoc.TCP*

VM_5 presentation layer machine (*presentation towards network language level*) *prot.HTTP or others*

VM_6 user program level (*user* programming language level)

Network/Transport/ Presentation level protocols

- *Network/Transport* Internet level protocol :

  TCP/IP (Transmission Control Protocol/Internet Protocol).

- *Transport/Presentation* level protocols.
  - Hypertext Transfer Protocol (HTTP) – for the transmission of information via the WEB
  - Simple Mail Transfer Protocol (SMTP) – for the management of electronic mail messages
  - Network News Transfer Protocol (NNTP) – for the management of discussion groups
  - File Transfer Protocol (FTP) - for the transfer of files between remote machines

Vertical description of the network

platform (hardware-part) =

(VM_1, VM_2)

platform (system-part (software)) =

(VM_3 , VM_4)

User (software) workload=

(VM_5 , VM_6)

***Vertical* description of the network functions**

Machine VM_i

is

- set of resources R_i

- Language for their use L_i

Vertical description of the network functions

Each primitive (instruction or command) of Language L_i

is created by a program written in the L_{i-1} Language of VM_{i-1} machine

Vertical description

- VM_1 physical layer (in LAN & GW)
  - resource R_1 = physical components (electrical, electronic), connections among these etc.
  - language L_1 = undefined
    in that level 1 is the only one where the functionalities aren't further emulated

Vertical description

- VM_2 LLC/MAC Layer (in LAN & GW) machine

–R_2 resources = internal registers, logic-arithmetic operators, sequencer, controllers, shifters, transfer networks, etc.

–L_2 language = microprogram language
  - uses resource R_2
  - its (micro)instructions or (micro)commands are directly interpreted by the physical machine VM_1

Vertical description

- VM_3 IP Layer (in HOST)
- – resources R_3 = memory locations (main, secondary), addressable registers of processors (main and peripheral or I/O), logic-arithmetic unit, control etc.

  – language L_3 = base language of HOST (machine language) in binary
    - uses R3 resources
    - each instruction or command written in L_3 is implemented by a microprogram written with (micro)instructions of language L_2 of VM_2

Vertical description

- VM_4 Transport Layer (in HOST)

  – resources R_4 = physical *spaces* of memory (main and secondary), *physical* processors as a whole (main, peripheral) etc.,

  –L_4 language = operating system language of HOST for programming *assignment procedures* R_4 resources according to the need which begin from the higher level:

  scheduler to govern memory spaces
  dispatcher to govern processors
  supervisor to govern peripheral units etc.

Vertical description

- VM_5 Presentation Layer (in HOST)

  – R_5 resources = *logic* spaces of memory (files, databases etc.), *logic* processors which interpret instructions in language L_4

  – L_5 language = symbolic programming language, standard language (C, C^{++}, Java etc. + HTML *for web documents description*) developed by having assembler instructions at a internal *sub-level* to VM_5 itself.

Vertical description

- VM_6 Application Layer (in HOST)
  - R_6 = still *logic processors* and *logic* spaces of memory, but at a more abstract level of level 4 logics, in that the processors now interpret commands written in language L5
  - L_6 language = higher level language, or a synthetic language (menu commands etc.)
  - logic memory spaces gather all that has been created by user programs at this level, thus file icons, folders etc.

Internet service (WPA program from HA (VM$_5$ / VM$_6$ level) towards HB)

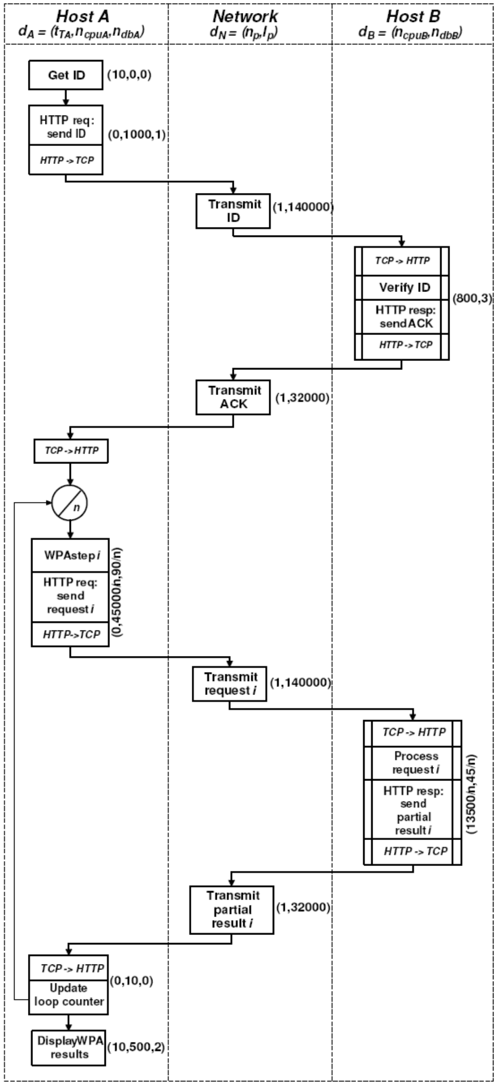
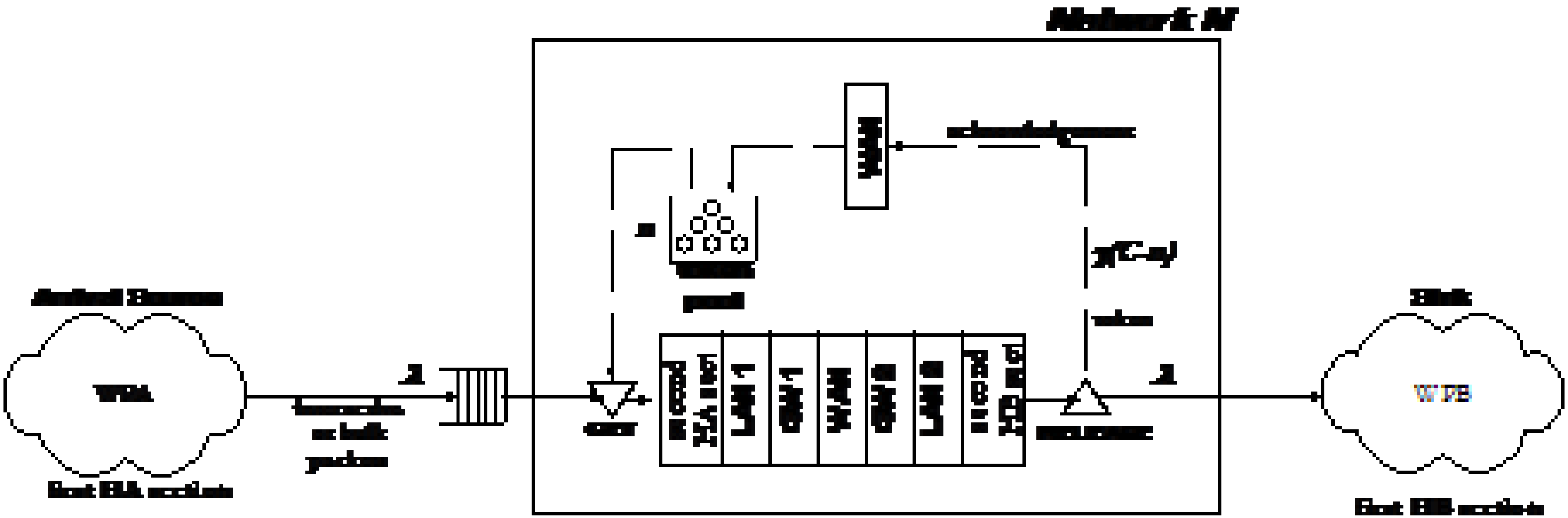


Fig. 3. Software model for the typical web application

WPA service towards platform



Informal platform model, WPA&WPB services and protocols

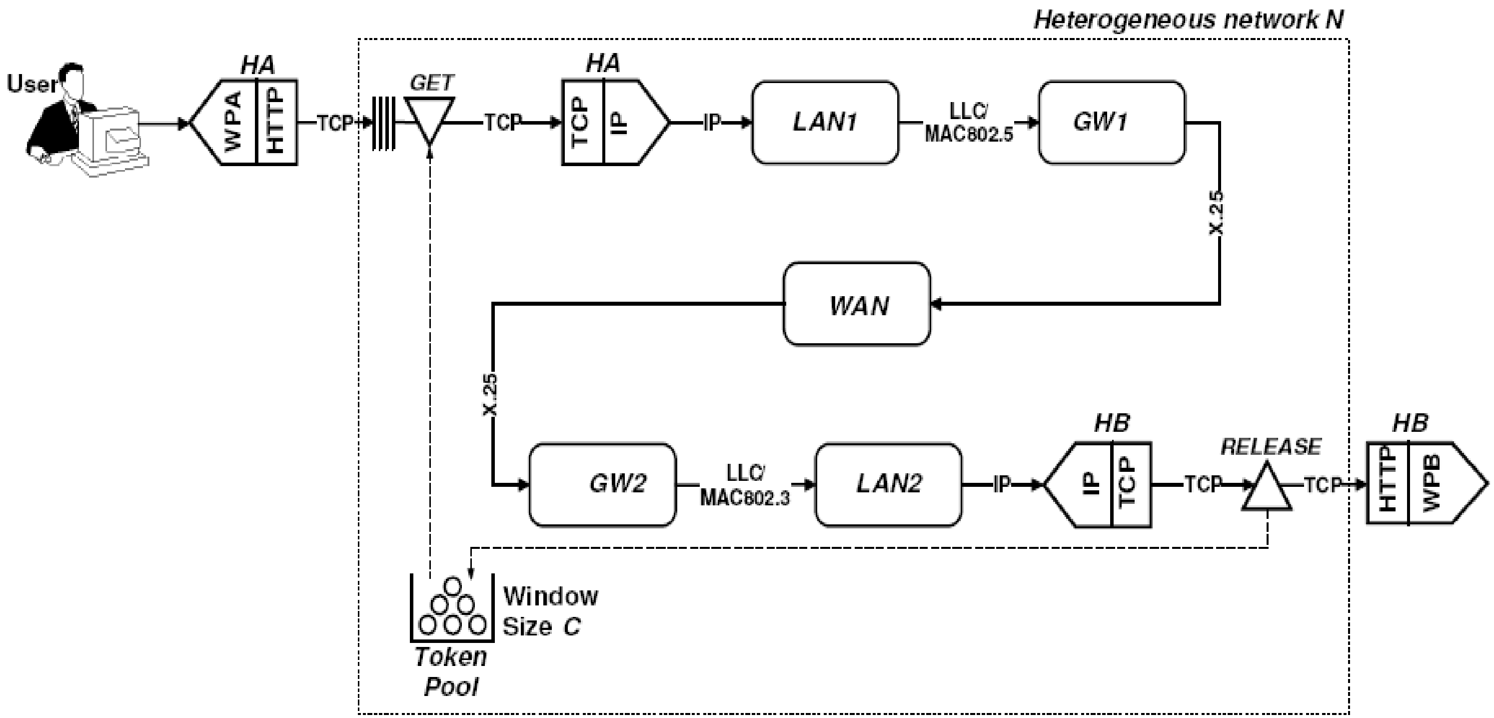
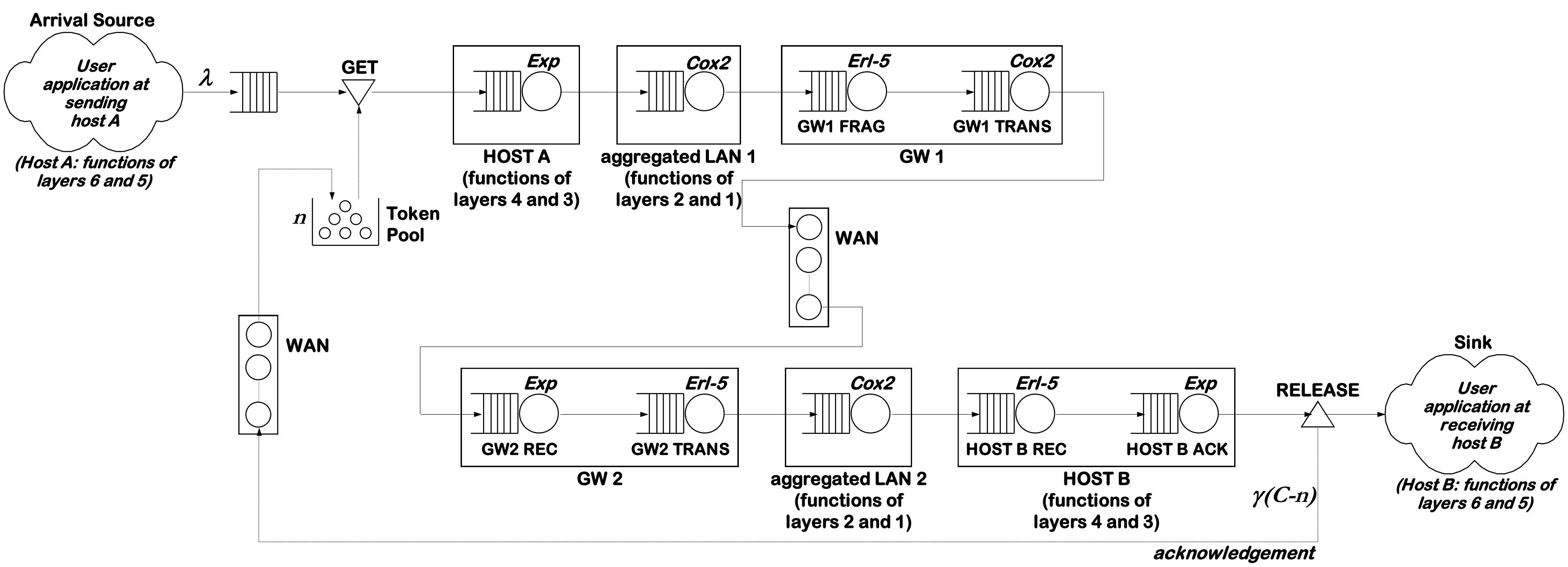


Fig. 2. Detailed view of the packet flow in the system platform when transferring data from HA to HB

Formal platform model, WPA&WPB services and protocols



Calculations on formal models: response time

- A is an access node (host)
- *response time* seen by A =

  = time from the instant in which a user in A launches a request to another resource R of the distributed imp., to the instant in which the response from R arrives to the user

= from A to R journey time

+ from R to A return time

end-to-end delay

- *If R = other host B*

  where another user is the one which wants to communicate with A (or vice versa).

  only interests the message journey time from A to B (or vice versa)

- This time is called **end-to-end delay** =

  = delay from one end to another of the platform

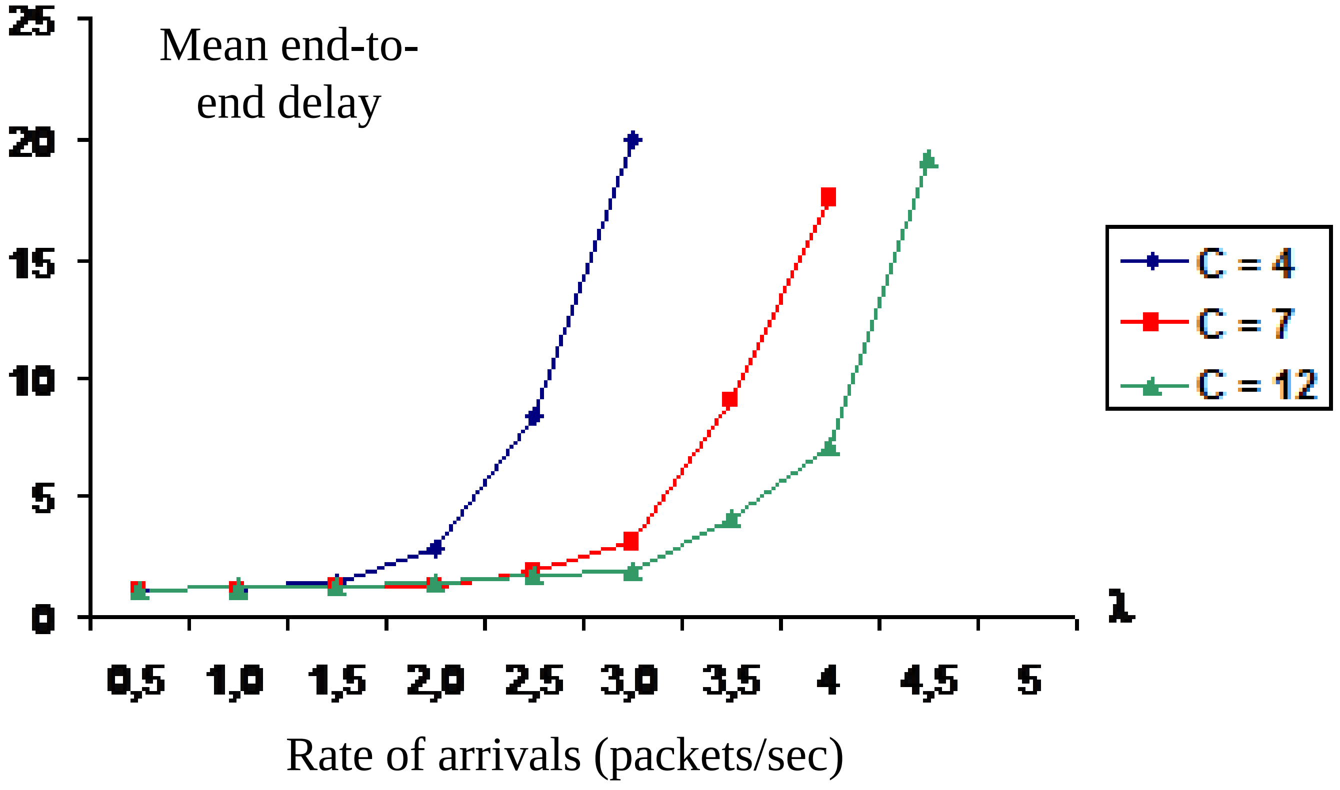
Calculations on the formal model: *network throughput*

- A is an access node (host)
- B is a node (host) which interacts with A

throughput of the network =

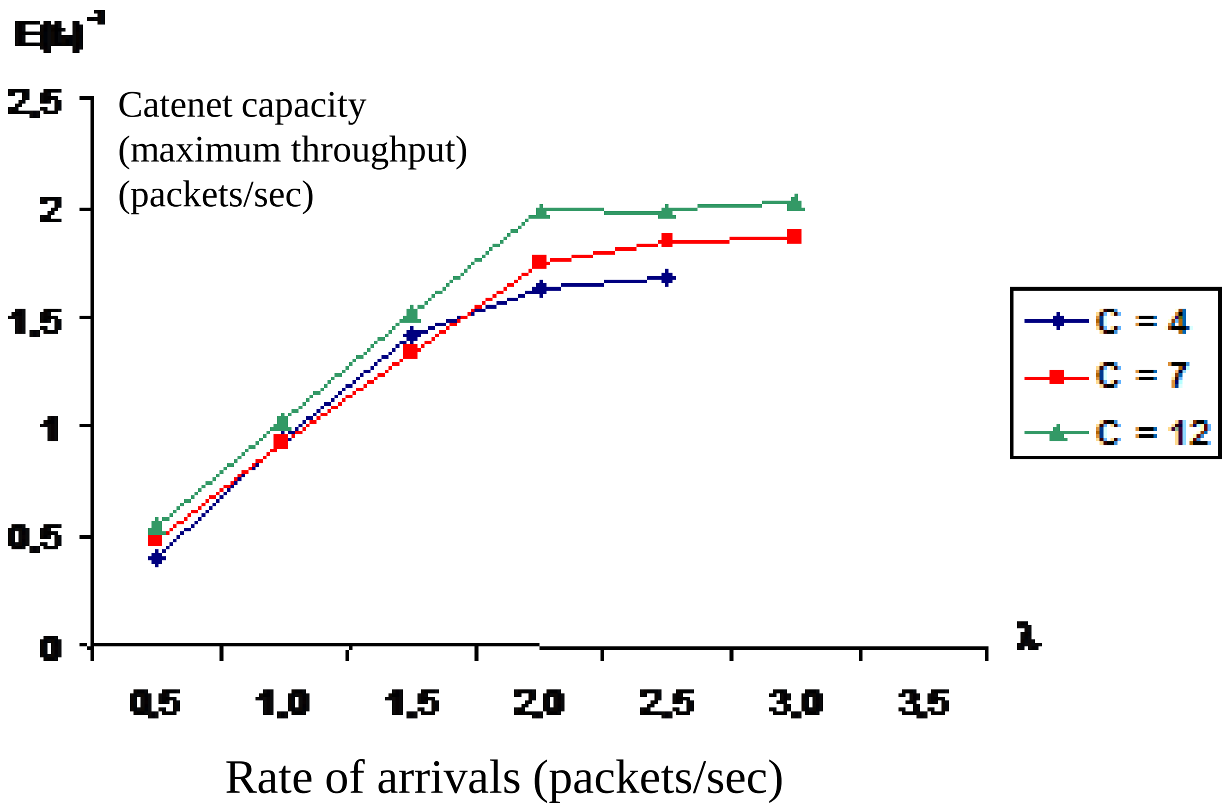
= Number of packets per unit of time that transit via the network

Example of results from the formal model
prediction of the window size effect on the end-to-end delay



Example of results from the formal model

prediction of the window size effect on the system capacity



Example of results from the formal model

(predict the execution time E(t) of the internet service)

study of the processing power effect of HA and HB and the workload platform ***n***

- The average execution time E(t) is **the sum** of the response times the application **spends in the various** service centers visited by application tasks.

- In other words E(t) is the global, or end-to-end response time *(in both directions)*.

- We assume that the designer is interested in **predicting the effect on E(t)** of concentrating the processing power in HA or HB for a given capacity of the network N.

- Predict E(t)

  versus 7 different combinations of

  the processing power Ca and Cb of HA and HB,

(predict the internet service execution time)

*study of the processing power effect of HA and HB and the workload platform **n***

- Various values of the reference processing power C were experimented up to finding out the two most significant cases:

  a) *the network is not the system bottleneck*

  b) *the network is the system bottleneck*

- For a network N with standard logical capacity values of various centers from HA to LAN1 to WAN to etc…to HB, it was found that

  $C = 10^3$ statements/sec

  and

  $C = 10^5$ statements/sec

- were the values of the reference processing power C to obtain the cases a) and b) respectively.

(predict the internet service execution time)

study of the processing power effect of HA and HB and the workload platform ***n***

- Seven different combinations of HA and HB processing power:

  - (1) Ca = 5C Cb = C
  - (2) Ca = 3C Cb = C
  - (3) Ca = 2C Cb = C
  - (4) Ca = Cb = C
  - (5) Ca = C Cb = 2C
  - (6) Ca = C Cb = 3C
  - (7) Ca = C Cb = 5C

Note:

When C = 10^3 *statements/sec (network IS NOT the system bottleneck)*

When C = 10^5 *statements/sec (network IS the system bottleneck)*

(predict the internet service execution time)
*study of the processing power effect of HA and HB and the workload platform **n***

- The value of the looping factor **n** gives the network workload (platform workload)
- Indeed, the demand vectors of all nodes included in the loop are divided by n, so that whichever is the value of n, the workload on HA and HB remains unchanged.
- On the contrary, the only workload that changes is the network workload, which is taken **n** times.
- The value **n = 9** was chosen as the lowest value in order to obtain a significant network workload.

(predict the internet service execution time)

study of the processing power effect of HA and HB and the workload platform ***n***

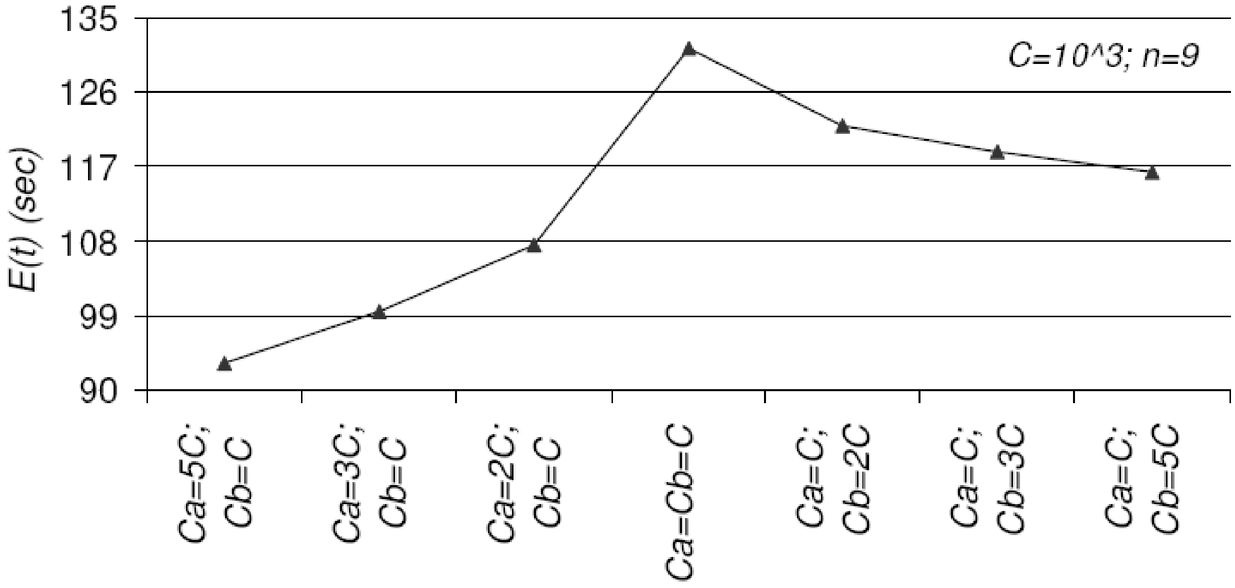


Fig. 9. $E(t)$ versus combinations of C_a and C_b for $C = 10^3$ statements/sec (network is not the system bottleneck)

(predict the internet service execution time)

*study of the processing power effect of HA and HB and the workload platform **n***

- When the network is NOT the system bottleneck, the host processing power has visible effects on E(t):
- *E(t) progressively decreases from its highest value E(t) = 131 sec to significant lower values when the processing power is concentrated on either HA (left side of the curve) or HB (right side of the curve).*
- *In the former case a decrease from E(t) = 131 to E(t) = 93 sec is seen with a 29% decrease.*
- *In the latter one, a decrease to E(t) = 116 is recognized, with a 11.5% decrease*

(predict the internet service execution time)

study of the processing power effect of HA and HB and the workload platform ***n***

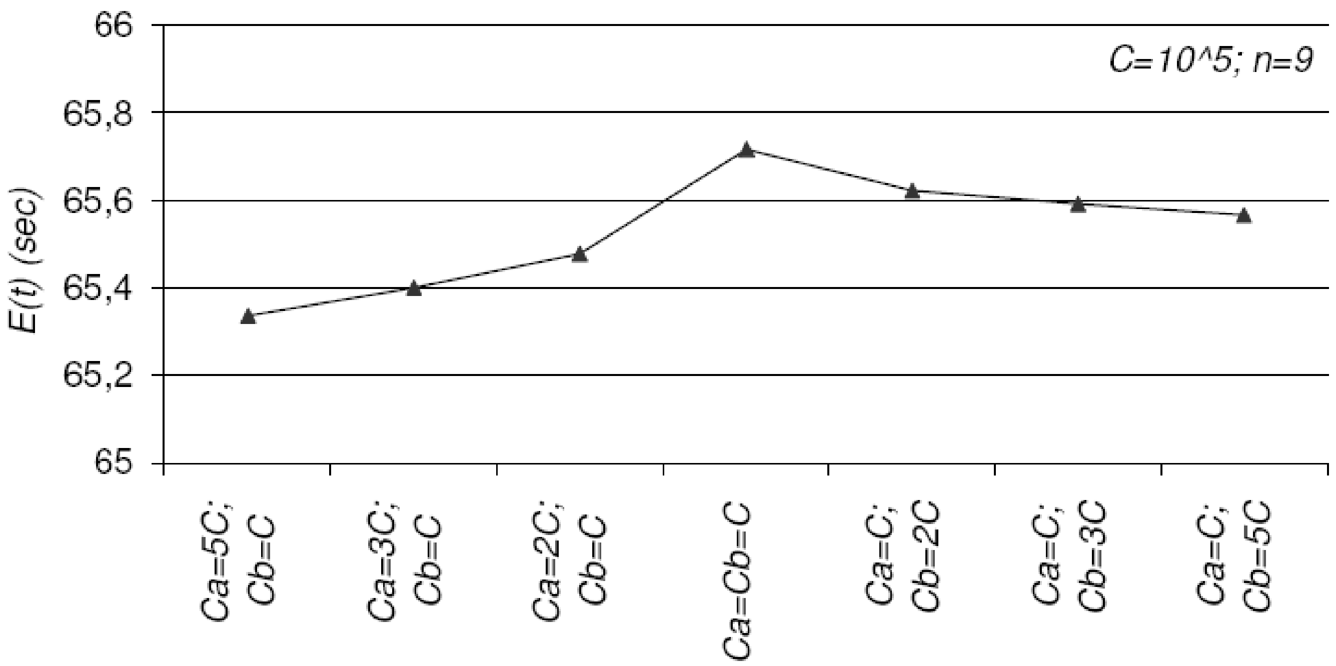


Fig. 10. $E(t)$ versus combinations of C_a and C_b for $C = 10^5$ statements/sec (network is the system bottleneck)

(predict the internet service execution time)

*study of the processing power effect of HA and HB and the workload platform **n***

- Instead, when the network IS the system bottleneck:

  there is no impact on E(t) of the processing power concentration on either HA or HB

  Since the network is largely responsible for the delays

Conclusions

- The dependence of the quality of the ***internet service*** on the
  - ***processing rates* of the** network **versus** the hosts
  - and **of the hosts**
- shows that **internet services** such as:
  - *distributed information retrieval*
  - *distributed interactive video*
  - *mobile telecommunication services*
  - *industrial process control*
  - *remote network management*
  - *network-based cooperative work*
  - *electronic commerce B2C , B2B, C2C*
  - *etc*
- can benefit from **formal model** studies from the

  early phases of **the internet development lifecycle**

  instead of using **fix it later** or **fly fix fly** approaches

QoS platform control

Model-driven management of QoS of platforms and internet services

QoS framework

- ISO/IEC 13236-1998 International Standard
- Structured collection of concepts to describe the Quality of Service (QoS) of IT systems
- Intended to assist those that produce specification and design of IT systems and those that define the communication services

- QoS characteristic of the network

  a quantified aspect of QoS, for example time delay, capacity, accuracy etc

- QoS requirement

  the user expectation of the QoS, for example, is the expectation that the time for a specific service (e.g. downloading a stream of data) must not exceed a specified value

Considered QoS characteristics

- *capacity-related characteristic*

  network *throughput,* i.e. number of packets per time unit delivered from source to sink through the network

- *time-related characteristic*

  network *end-to-end delay,* i.e. the time for a packet to get across network

Example of QoS requirements

- *capacity-related* characteristic
  - **E1**: the time to download a stream of data of *x* kByte of length from sending host A to receiving host B should not exceed *y* sec.
- *time-related* characteristic
  - **E2**: the time it takes host B to receive a command of one packet length sent from host A should not exceed *z* sec.

QoS management/maintenance

- network QoS management

  all activities designed to assist in satisfying one or more QoS requirements

- operational phase of QoS management

  management activities intended to honor the agreements on the QoS to be delivered

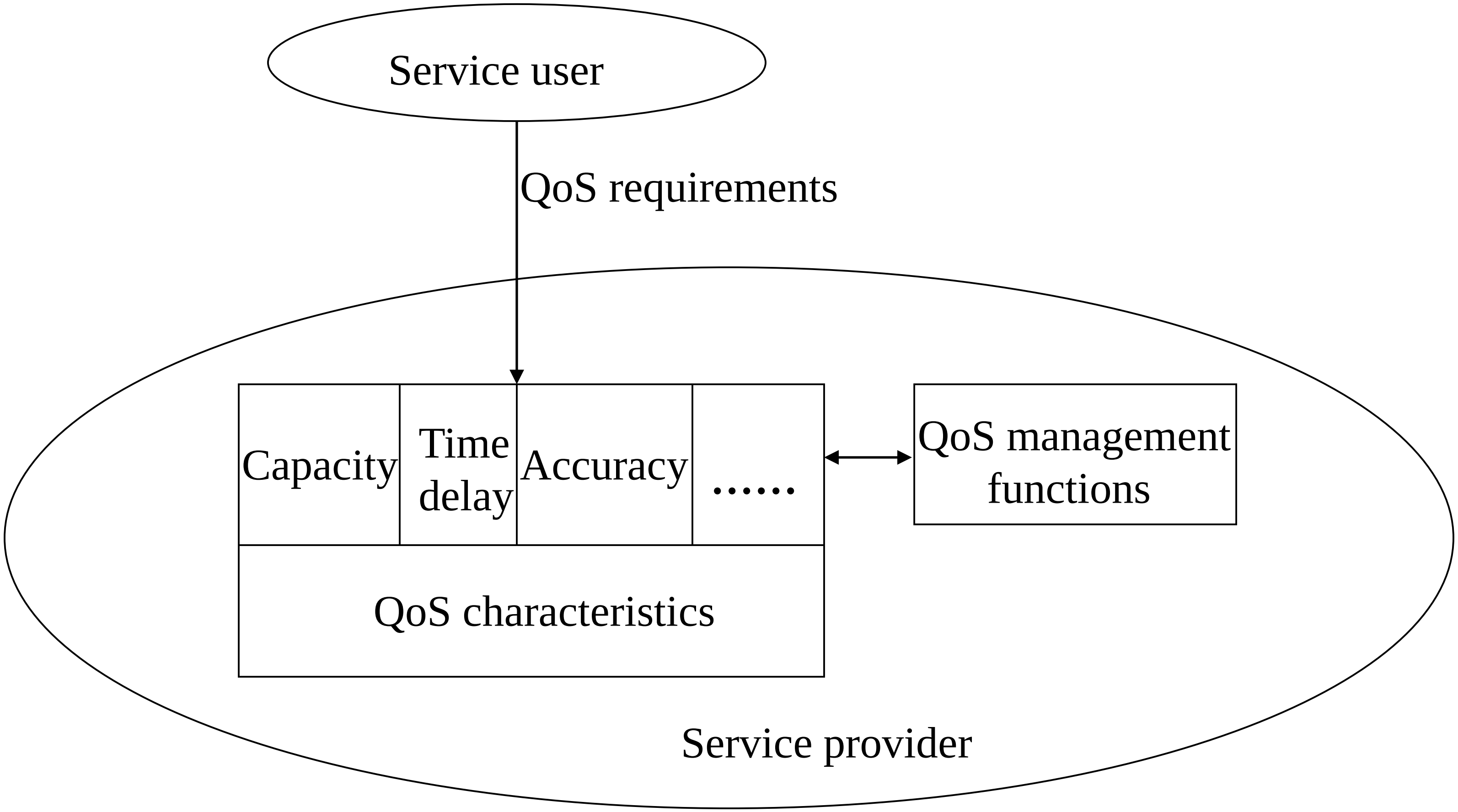
- network QoS maintenance

  the activity intended to maintain QoS to acceptable levels (tuning activity)

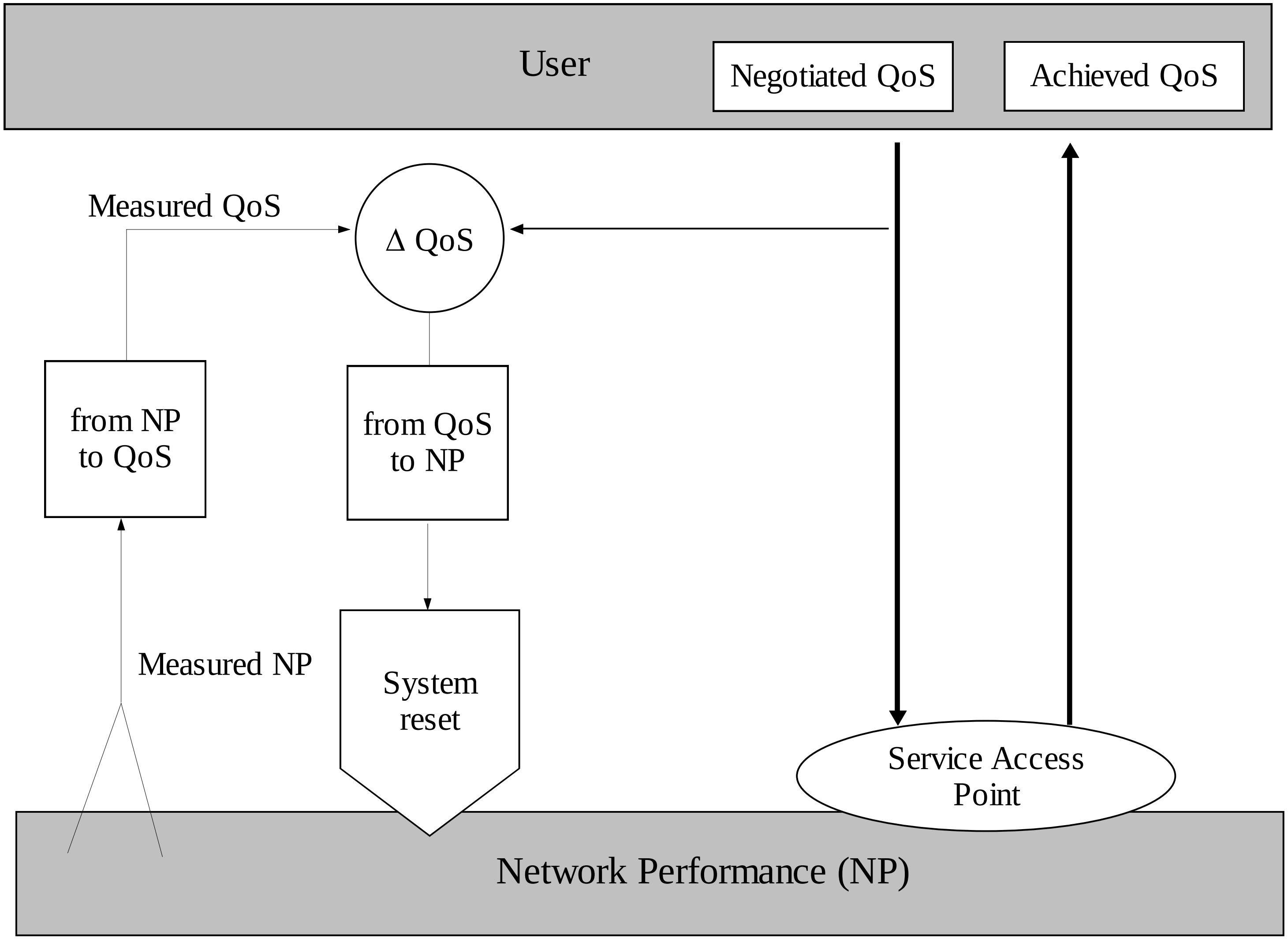
QoS management/maintenance

- Network tuning
  - a compensatory adjustment of the network operation
  - directed by an efficient performance model, an on-line approach to system evaluation

Relationships between QoS concepts



Network QoS tuning



Time-efficiency of the performance model

- *model-based tuning* = identify the network variables to reset in a short time

time-efficiency essential

to conduct the *decision process* in due time

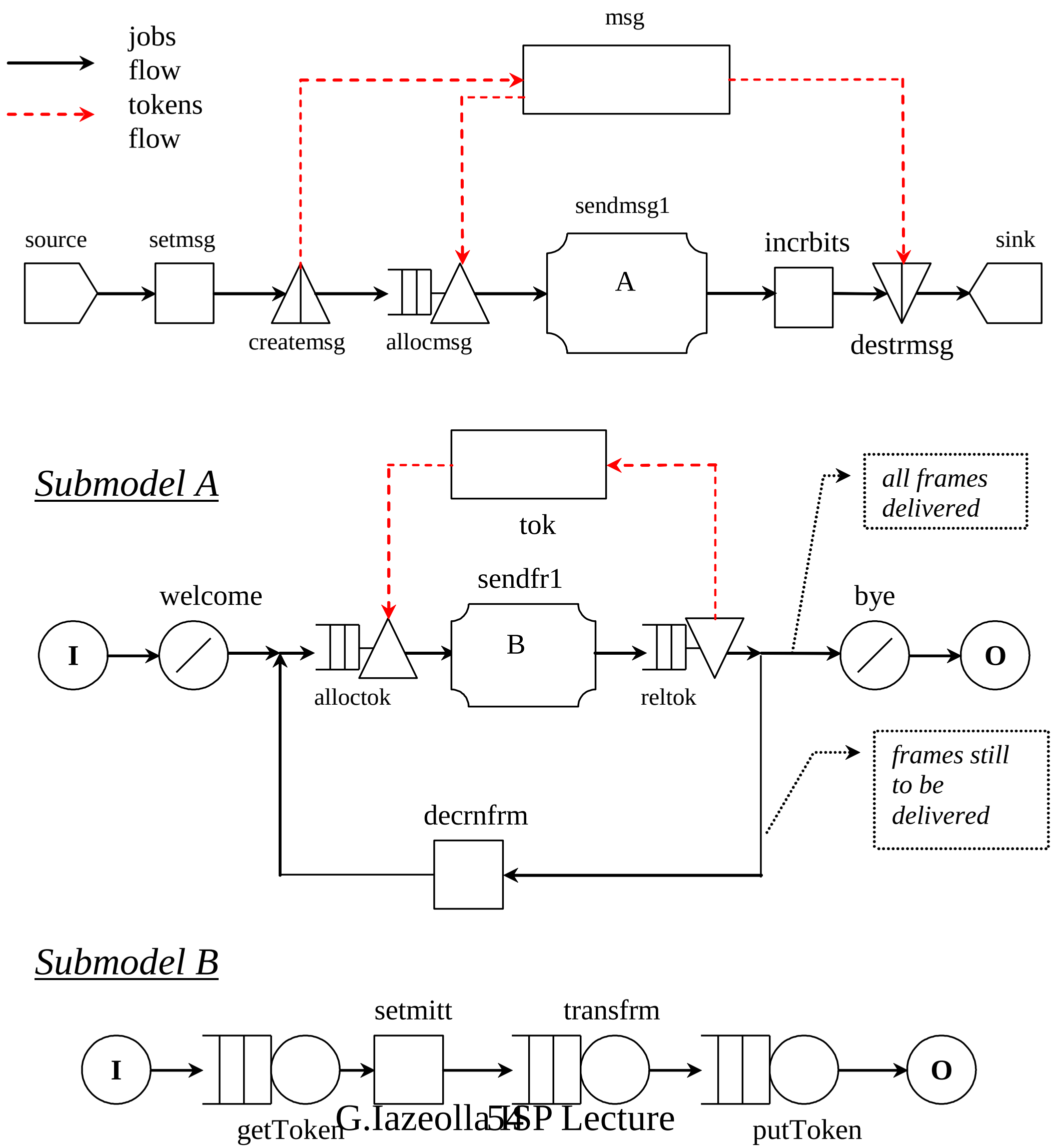
Time-efficiency objective

Reduce model evaluation time from about 30 hours of the brute-force approach to a few seconds of the efficient approach

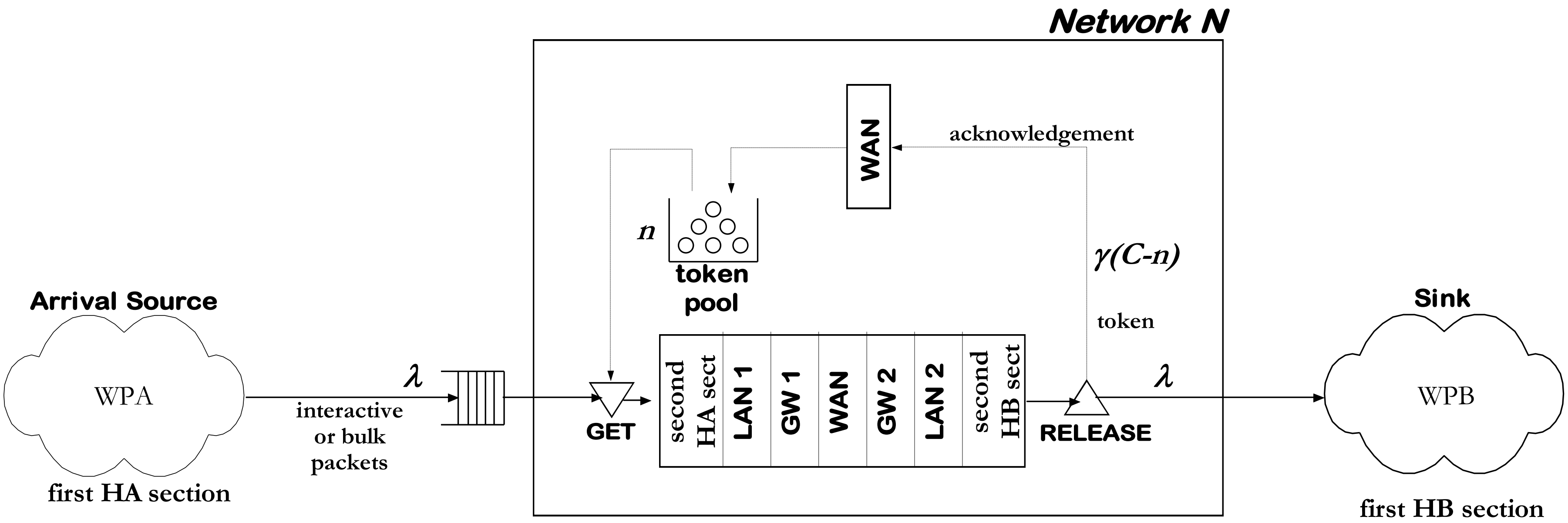
Efficient evaluation method

- Hierarchical hybrid approach based on decomposition and aggregation
- Based on three abstraction levels

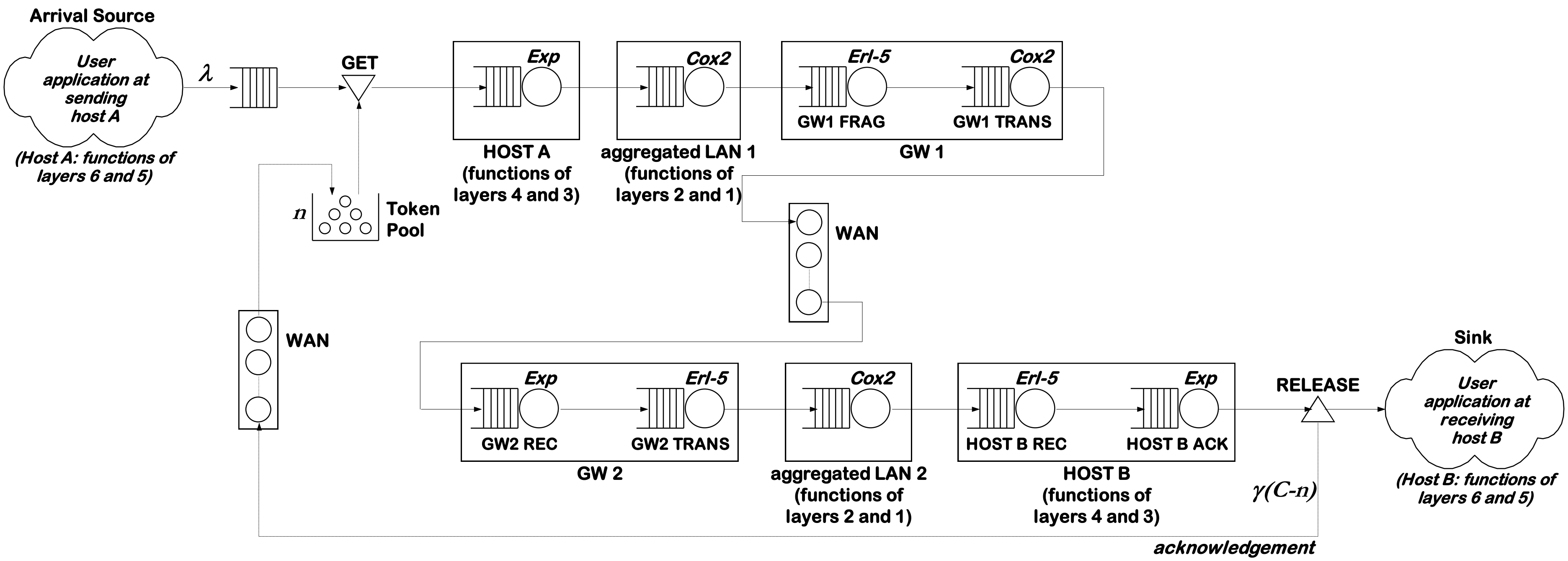
Level-1 Model (part of)



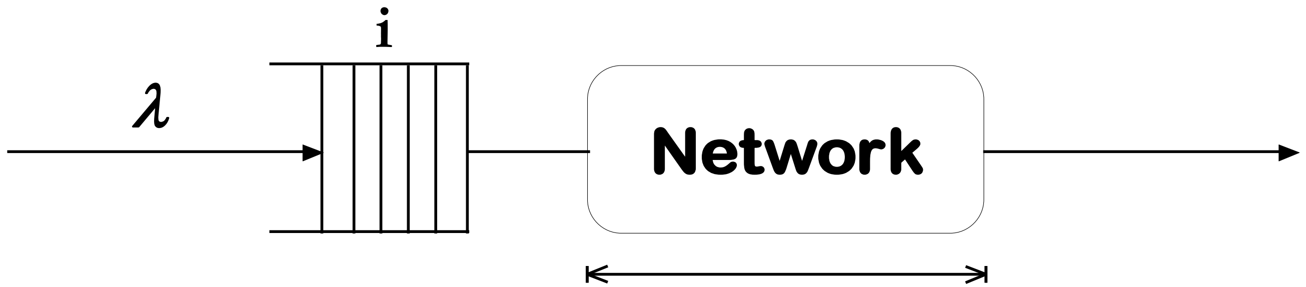
Level-2 Model



Level-2 Model (expanded view)



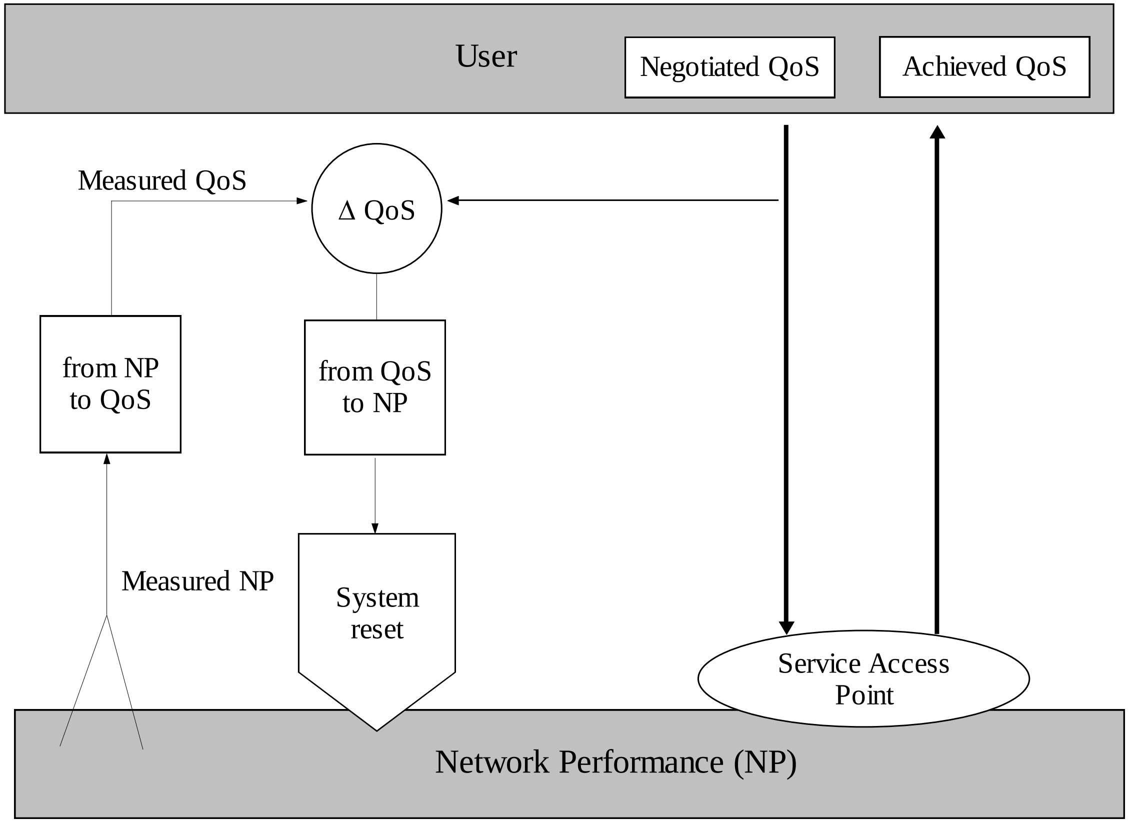
Level-3 Model



$E(t_s)$ = function of ack throughput (γ(C-n))

$$E(t_s) = 1 / \gamma(i) \quad \text{for} \quad 0 \leq i \leq C$$
$$= 1 / \gamma(C) \quad \text{for} \quad i > C$$

Network QoS tuning



a) measurement of the system performance (SP) characteristics, e.g. mean end-to-end delay or mean network capacity

b) translation to QoS values

c) calculation of QoS difference

d) identification of QoS related components

e) modification of parameters of identified components and system reset

Decision process of tuning operations

The tuning decision process, i.e.:

d) identification of QoS related components

e) modification of parameters of identified components and system reset

… is performance-model driven

Identification of QoS related components (step 1)

- QoS user requirement:
  - the time for downloading a stream of data must not exceed a specified value
- related QoS characteristics:
  - *end-to-end delay*
  - network *capacity*

Identification of QoS related components (step 2)

- packet arrival process
- packet acknowledgement process

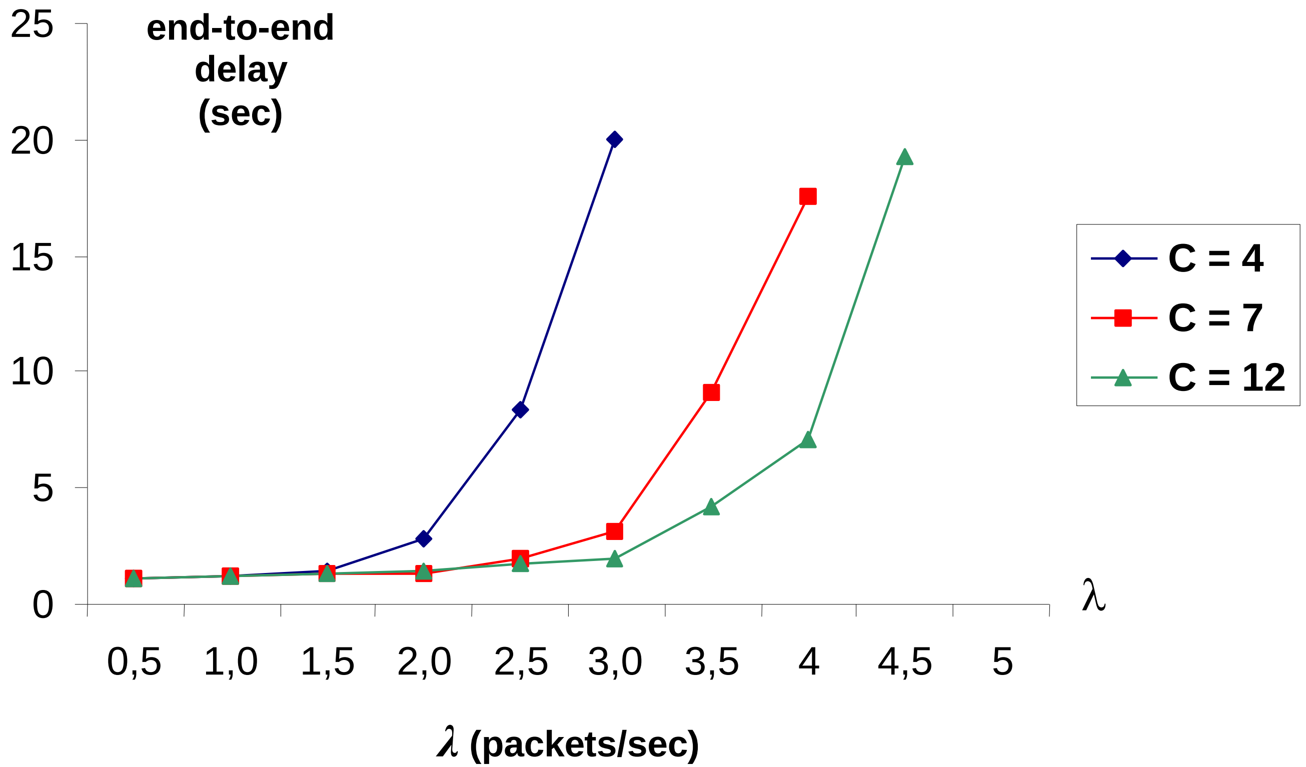
Parameters of identified components

- arrival rate λ of user application packets
- network window size C

Network reset operations

Illustration of the effect of variables reset on the mean end-to-end delay and on the network capacity

Effect on *end-to-end* delay



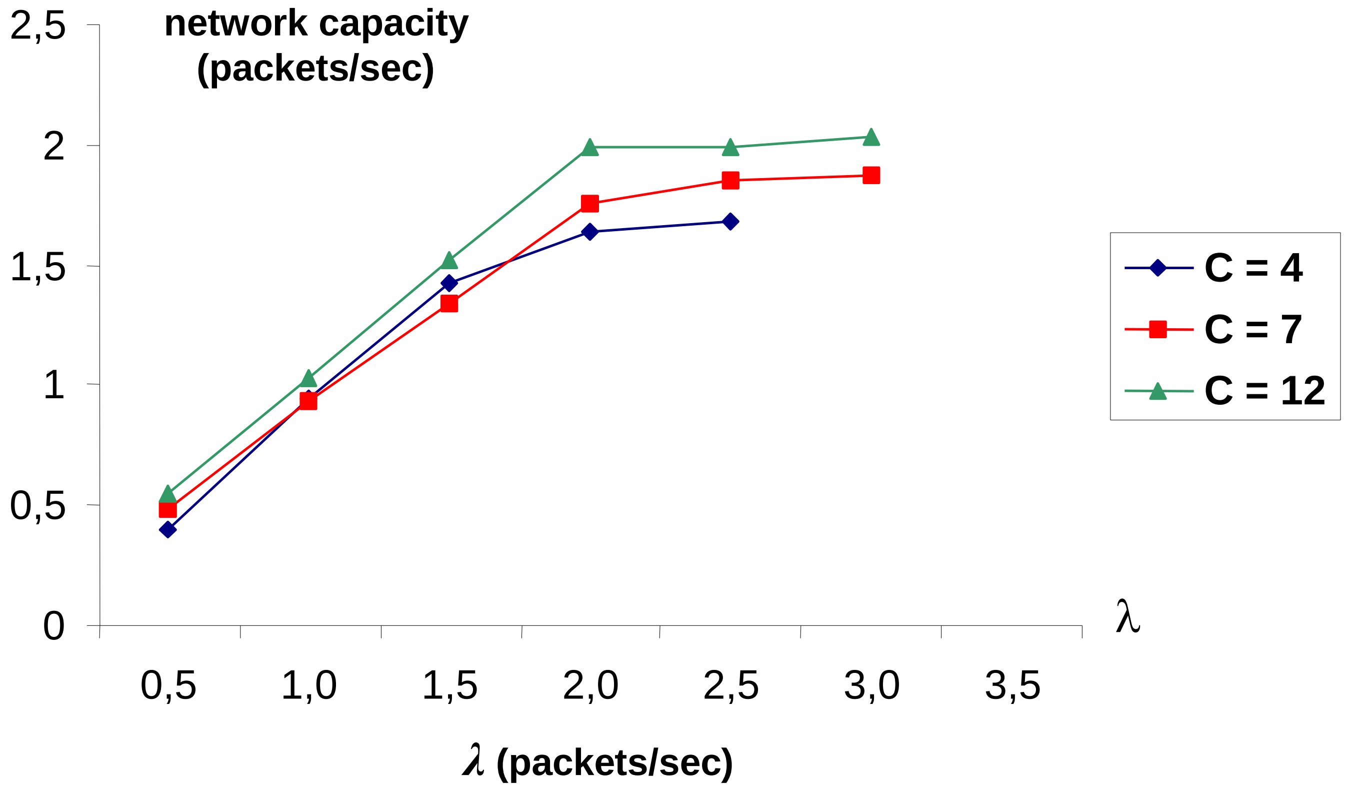
Mean end-to-end delay versus λ

Effect on *end-to-end* delay (2)

- Assume we are in the case of the QoS requirement E2, and that the negotiated QoS is a time of no more than 15 seconds to deliver a command of one packet length to B.
- By using the model, it is possible to find the values of the parameters λ or C that guarantee a mean end-to-end delay lower than or equal to 15 seconds. By looking at the previous figure it is easily seen that:
  - for $\lambda \leq 2{,}8$ all values of the window size C (4, 7 or 12) can be chosen,
  - for $2{,}8 < \lambda \leq 3{,}8$ C=7 or C=12 can be chosen,
  - for $3{,}8 < \lambda \leq 4{,}3$ only C=12 can be chosen,

  while no values of C can guarantee the considered requirement if $\lambda > 4{,}3$.

Effect on *network capacity*



Network capacity versus λ

Effect on *network capacity* (2)

- Assume we are in the case of the QoS requirement E1, and that the negotiated QoS is a time of no more than 10 seconds to download a stream of data of 140 Kbytes. In other words, a throughput of no less than 14 Kbytes/sec, or 1,75 packets/sec for packets of 8 Kbytes length, is required.
- By using the model, it is possible to find the values of the parameters λ or C that guarantee a mean network throughput greater than or equal to 1,75 packets per second. By looking at previous figure we can easily see that for $1,75 \leq \lambda \leq 2$ only a window size C=12 can be chosen, while for $\lambda > 2$ C=7 can be chosen as well.

Conclusions

- Performance modeling is called to play an important role in the network QoS management activities
- To play such a role, however, some important improvements need to be achieved in the modeling research and application
  - Methods for QoS-oriented model production
  - Model capability in identifying QoS-related components and parameters
  - Model evaluation effectiveness for its on-line use in the tuning operation loop

Complete development of the formal model

Next lectures