



# Beyond Technology: The Missing Pieces for QoS Success

**SIGCOMM 2003 – RIPQoS Workshop**

L. Burgstahler, K. Dolzer, C. Hauser, J. Jähnert, S. Junghans, [C. Macián](#), W. Payer

Institute of Communication Networks and Computer Engineering

University of Stuttgart

{burgstahler,dolzer,hauser,jaehnert,junghans,macian,payer}@ikr.uni-stuttgart.de

August 27, 2003

- ❑ **Motivation**
  - ◆ What is QoS?
  - ◆ Why do we need QoS?
- ❑ **Current Approaches**
- ❑ **Critical Issues**
- ❑ **Possible Directions**
- ❑ **Conclusions**

# What is QoS?

**[ITU-T]:** The collective effect of service performance which determines the degree of satisfaction of a *user of the service*

**[IETF]:** A set of service requirements to be met by the network while transporting a flow

→ Service specific

→ Subjective perception

Responsibility: Access and core network(s) ...  
... but also end-user equipment

Provided by **network** or application

Scope: end-to-end

QoS is also availability, reliability, security, ...  
◆ not considered here

# Why do we need QoS?

## ❑ **Impossibility to provide satisfactory quality in Best Effort networks**

- ◆ In every environment
- ◆ For every service
- ◆ In a sustainable way

## ❑ **Potential solution: Overprovisioning**

- ◆ Problem: Severe resource scarcity in some (access) networks
- ◆ Problem: Costly investments necessary to overcome scarcity
- ◆ Problem: Physical limits

→ **Available resources do not satisfy all users' expectations**

→ **Some users willing to **pay more** for (somewhat) guaranteed resources**

→ **Main rationale for QoS: Money!**

# Which resources? How to allocate them?

## □ Resources

- ◆ Mainly bandwidth (and buffer space)
- ◆ Delay and jitter as boundary conditions
- ◆ Usually not independent

## □ Resource allocation

### ◆ Technical dimension

- Reservation, bandwidth overprovisioning
- Admission control, traffic shaping
- Traffic separation and scheduling

### ◆ Economic dimension

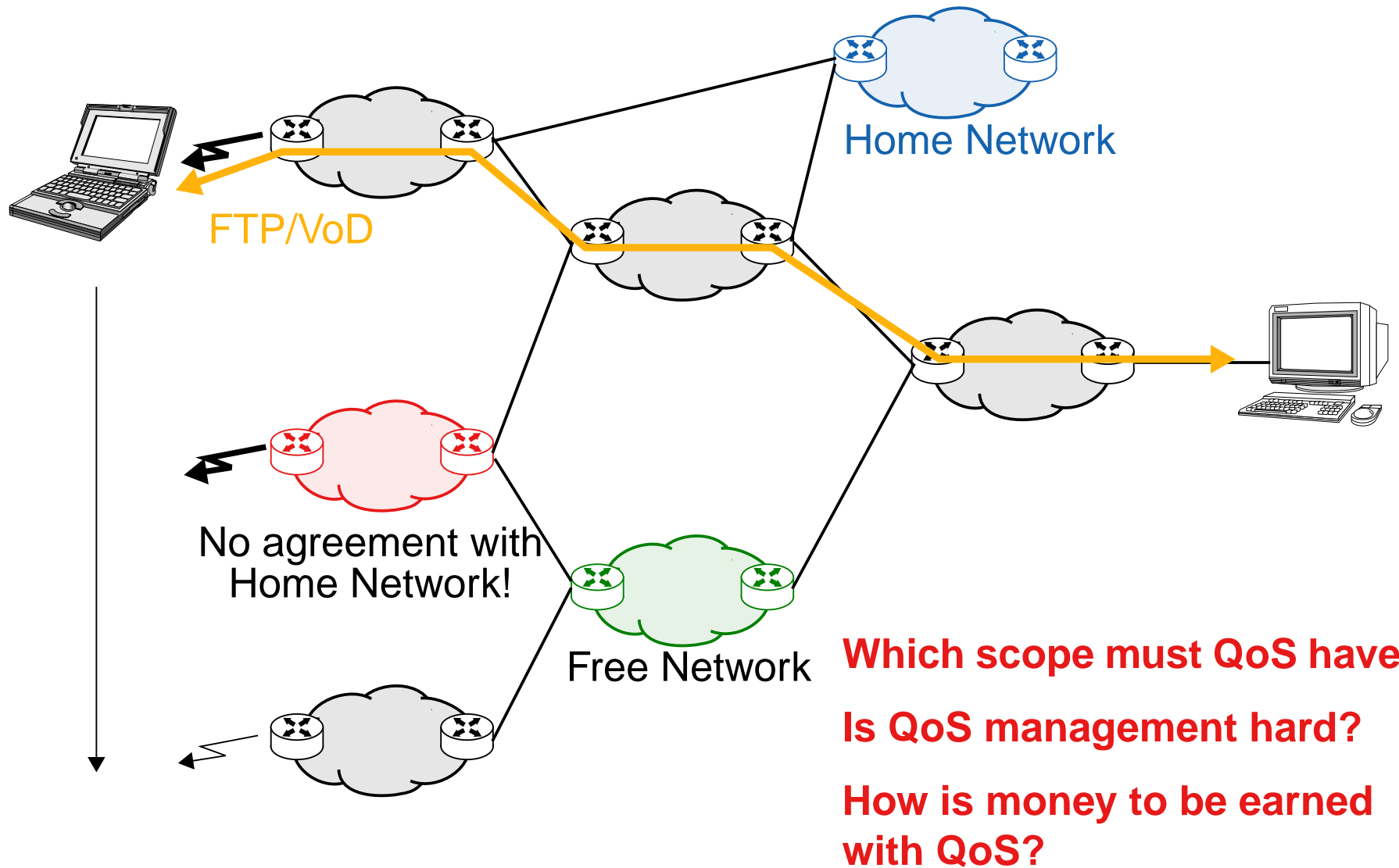
- Price differentiation
- Market segmentation
- Service bundling
- Auction

→ **QoS success will need a mixture of both**

# Current approaches

- ❑ **Stateful QoS: Application-triggered QoS signalling**
- ❑ **Stateless QoS: Decoupling of application and network load control**
- ❑ **Combined approach**
  - ◆ IntServ: Interaction of applications with the network
  - ◆ DiffServ: Management of admission to the network
- ❑ **Integration of (G)MPLS not foreseen**
  - ◆ Basis for DiffServ
- **Combined approach presents solution for**
  - ◆ Capability negotiation with peer
  - ◆ Extension of QoS architecture to the application layer
  - ◆ Integration of DiffServ in an end-to-end QoS model
  - ◆ Warning: Some issues still "work in progress"
  
- **QoS is in principle technically viable...**
  - ... but an integrated architecture is not yet foreseeable

# Critical Issues: Scenario



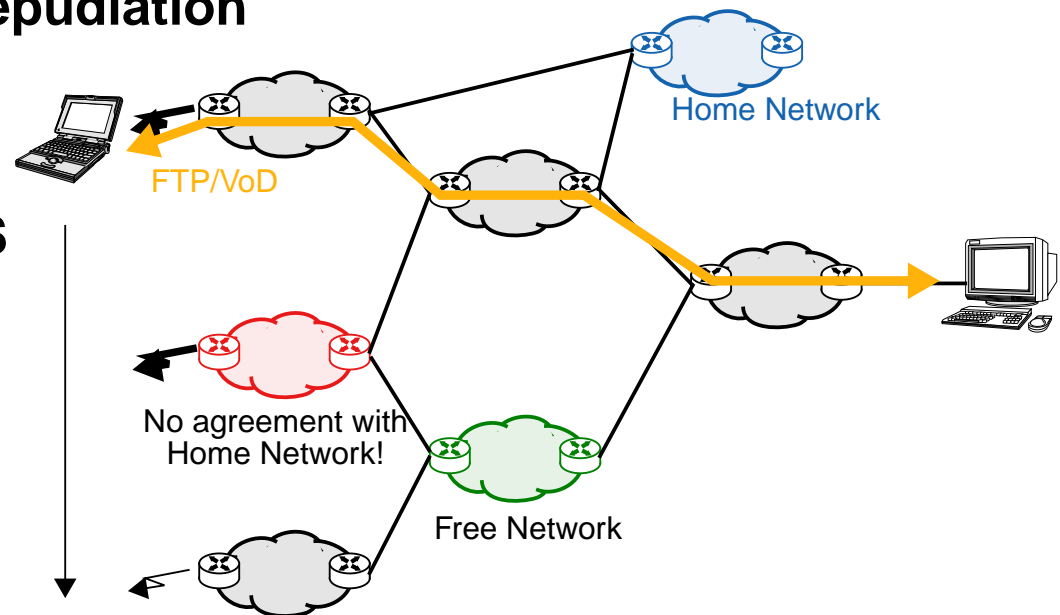
# Critical Issue: Which scope must QoS have?

- ❑ **Similar to connectivity: Either end-to-end or not at all**
- ❑ **Scope dependent on traffic pattern**
  - ◆ Small number of domains is responsible for a majority of the traffic
  - ◆ Source and destination are no direct neighbours
  - ◆ Within a few days traffic is exchanged with almost the complete Internet
  
- QoS needs to be available Internet-wide
- Any transaction implies domain interaction
- Complexity of management



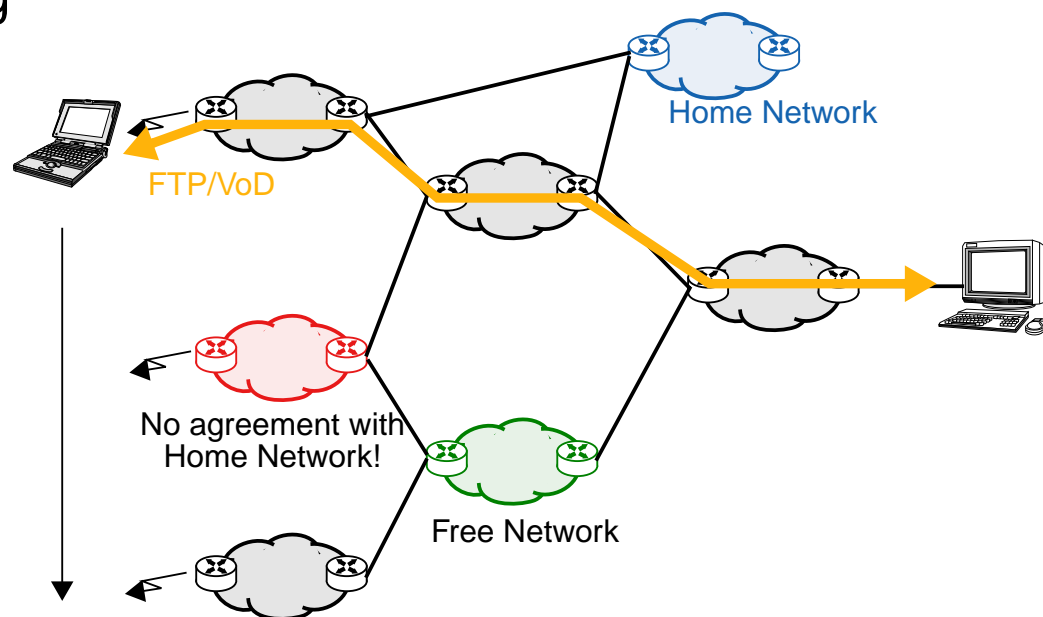
# Critical Issue: Why is QoS management hard?

- ❑ **Domain interaction**
  - ◆ Coherent traffic handling
  - ◆ Distribution of information
  - ◆ Bilateral contracts between domains
- ❑ **Traffic Characteristics**
  - ◆ Bidirectional property
  - ◆ Relationship between traffic flows
- ❑ **Responsibility, trust, non-repudiation**
- ❑ **Gathering of accounting information**
- ❑ **Influence of routing on QoS**



# Critical Issue: How is money to be earned?

- ❑ Networks provide data transport only → No added value!
- ❑ QoS could change that...
  - ... but there is no business model specifically for QoS
- ❑ Who pays whom for what?
  - ◆ Horizontal dimension: between domains
  - ◆ Vertical dimension: user - content provider- service operator- carrier
- ❑ User pays for usage of a service
  - ◆ Distance/"network crossing"
  - ◆ Volume/time
- ❑ Is QoS a standalone service?
  - ◆ Some services need a minimum quality
    - QoS is part of the service
    - What if Best Effort would be enough?



# What about accounting & billing?

❑ **No clear business model** → no accounting architecture

❑ **Additional intrinsic accounting problems**

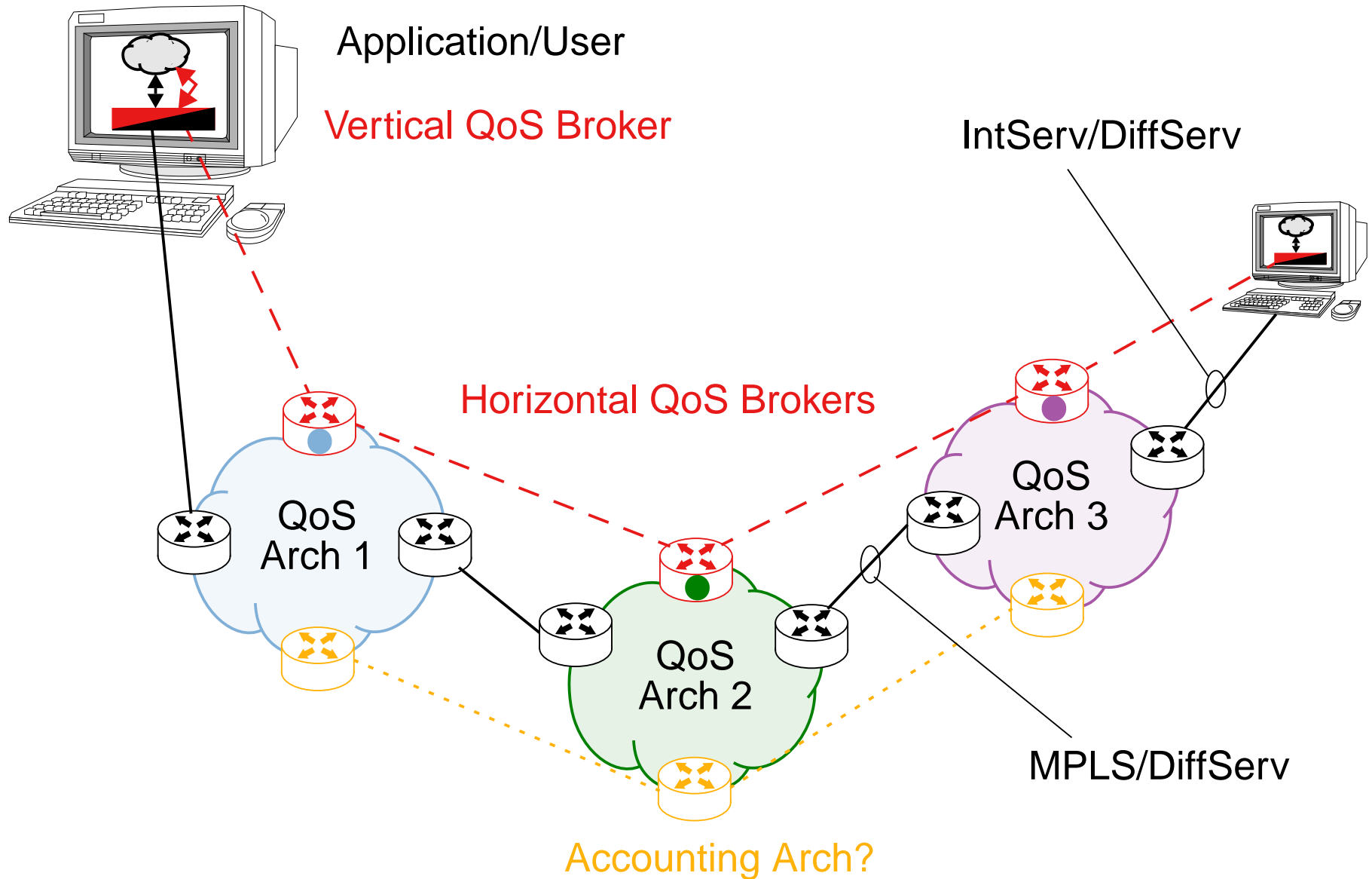
- ◆ Complexity through granularity
- ◆ Scalability
- ◆ Trust
- ◆ Cost
  - Up to 50% of all operating costs!

→ **Work in progress**

→ **No accounting & billing architecture** → No money → No QoS!

# A possible direction: QoS Brokers

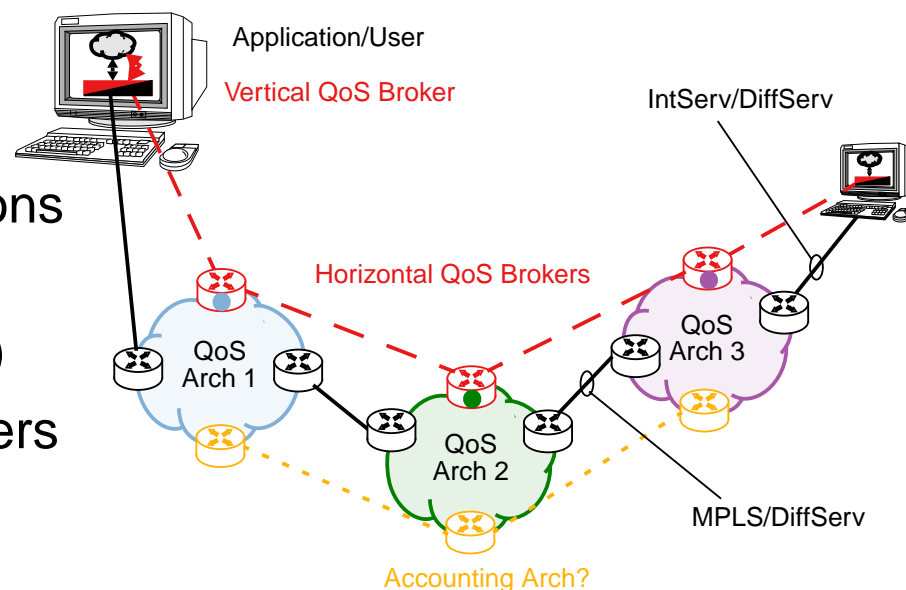
(I)



- ❑ **Missing Link: Automated, dynamic QoS management architecture**
  - ◆ No solution for business model or accounting architecture
- ❑ **Will applications and networks cooperate for joint delivery of QoS?**
  - ◆ Little probability of a unified network interface
  - ◆ Different QoS architectures: Mapping problem between application requirements and network capabilities
  - ◆ Application often not able to communicate quality requirements

## → QoS-broker/proxy/middleware

- ◆ Vertical and horizontal
- ◆ Terminates signaling with the different networks and applications
  - Implies willingness to cooperate
- ◆ Configured by the user (profiles)
- ◆ Includes non-technical parameters
  - Like price



# Other possible directions

## ❑ Market consolidation

- ◆ Reduces trust and responsibility issues
- ◆ Simplifies QoS management
  - Like QoS mapping
  - Like QoS information dissemination
  - Like QoS e2e coherence
- ◆ Simplifies accounting
- But **still no business model**

## ❑ Regulation

- ◆ Treat data transport (and QoS) like any other basic commodity
- **No solution**, but serves as "motivator"

## ❑ Stagnation

- ◆ Wait until overcapacity solves most problems
- ◆ Ignore the rest
- **Give up on QoS**

# Conclusions

- ❑ **QoS is necessary as long as there is resource scarcity**
- ❑ **Critical requirements to develop a viable QoS architecture:**
  - ◆ Provided end-to-end (and Internet-wide)
  - ◆ Management burden low in the face of heterogeneity and diversity
  - ◆ Clear business model and the corresponding accounting & billing architecture
- ❑ **QoS broker is a nice idea, but last issue still unresolved**
- ❑ **Many technical pieces available or feasible, integration missing**
- ❑ **Mainly non-technical aspects brake the advent of QoS**
  - ◆ Achievement of the critical mass
  - ◆ Business model
    - Accounting & billing architecture
  - ◆ Trust & cooperation
    - end-to-end coherence and management architecture