Part 3: Quality-of-Service (QoS)

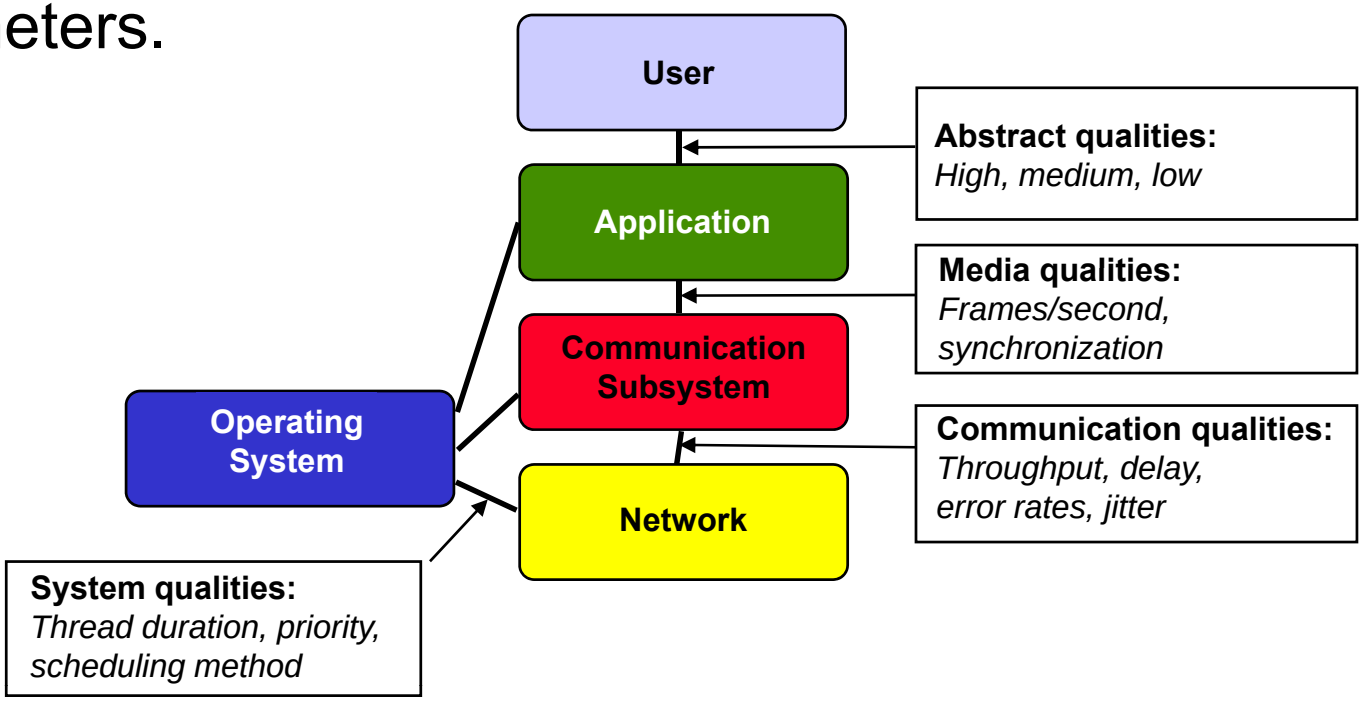
- QoS Basics
- QoS Modeling
- QoS Methods

QoS Basics

QoS – Classifications and Definitions
Type-of-Service (ToS)

Quality-of-Service (QoS) – Initial Example

- Different components of the communication architecture require distinct parameters.

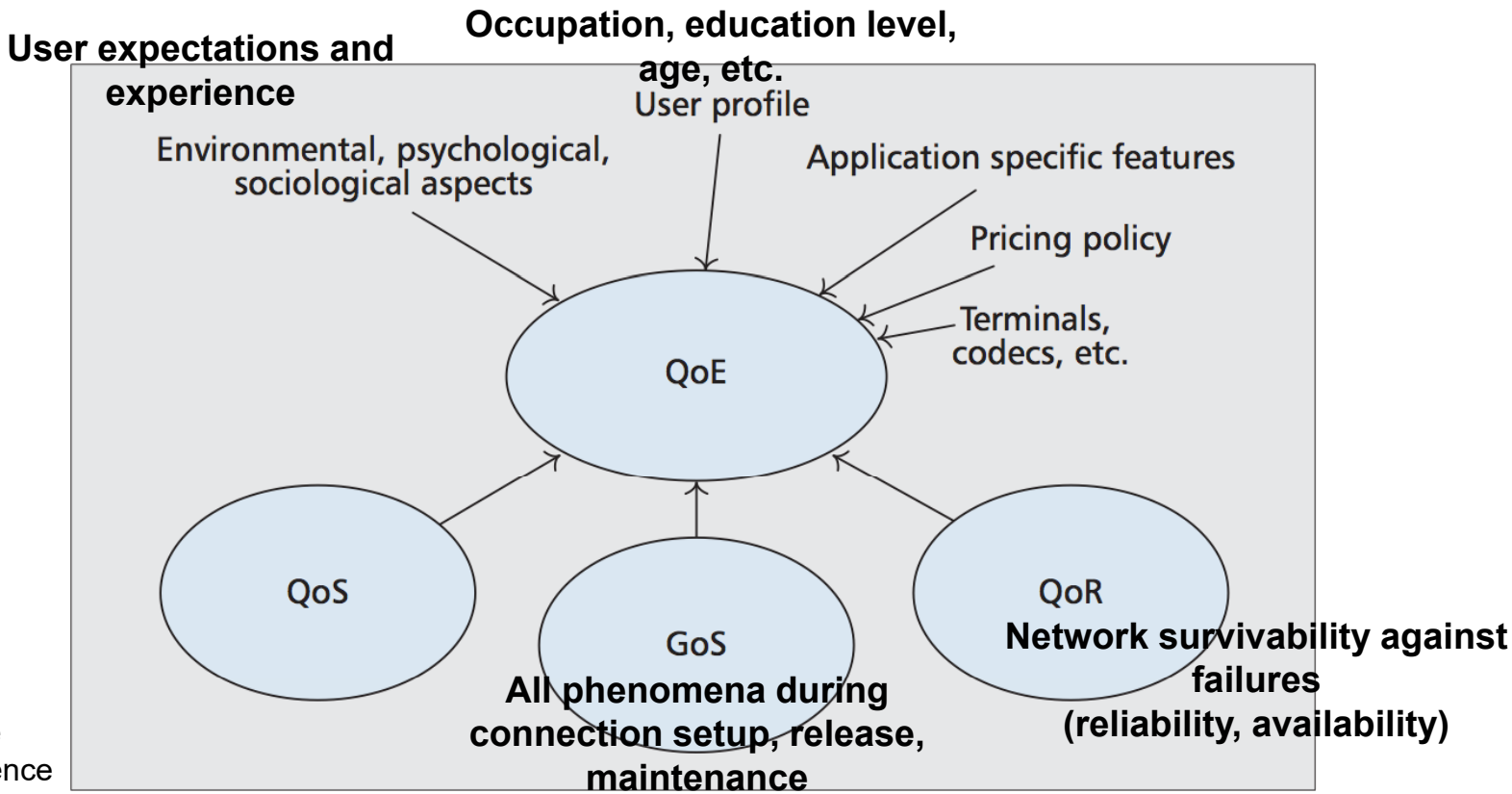


- A concept to describe service requirements and prerequisites
  - Quality-of-Service (QoS) models
    - Including amongst other things QoS parameter and QoS definitions

Quality-of-Experience (QoE)

- QoE encompasses the issue of a user's decision on retaining the service or giving it up
- Claimed factors influencing QoE:

R. Stankiewicz, P. Cholda, A. Jajszczyk: QoX: What is it really?
IEEE Communications Magazine, Vol. 49, No. 4, pp. 148-158, April 2011.



GoS: Grade of Service
QoR: Quality of Resilience

Problems with QoS and Definitions

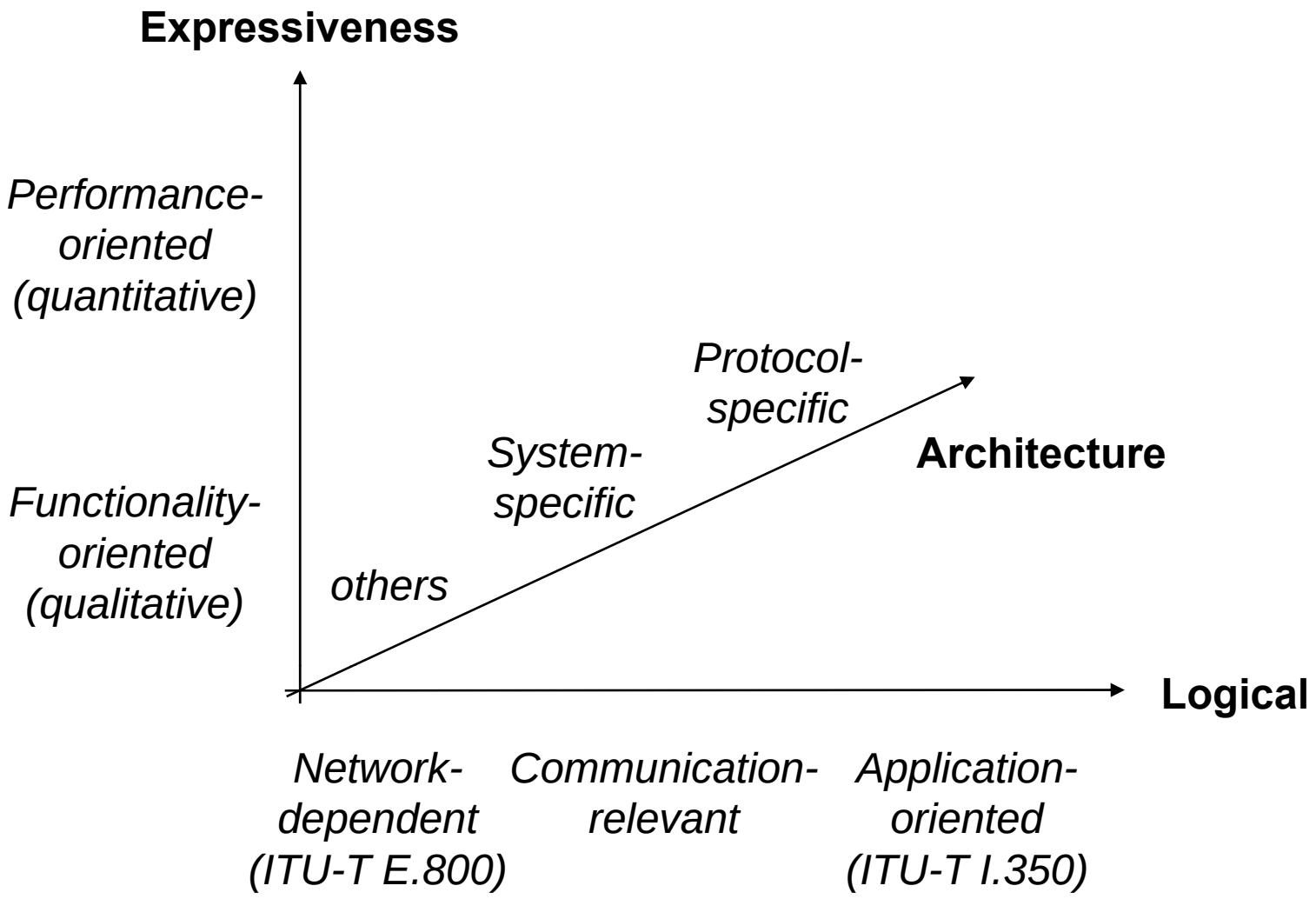
- Lack of overall framework
  - Distributed applications, communication system, network
- Incompleteness and inconsistency
  - Missing QoS parameters for different levels of abstraction
  - Service specifications do not provide a notion of QoS
  - Protocol definitions lack mechanisms in support of QoS
  - Internet versus ATM versus research
  - Traffic models do not comply with workload models
- Definitions
  - General approach: A concept for specifying requirements of communication services in terms of grade, excellence, or goodness.
  - ISO: A set of qualities related to the collective behavior of one or more objects.
    - QoS – Basic Framework, ISO/IEC JTC1/SC21 N9309 and BRM ODP
  - ITU-T: The collective effect of service performance which determines the degree of satisfaction of a user of the service
    - ITU-T E.800

Example IEEE 802.1Q/p

- Link-prioritization classification: MAC header.
  - 802.1D with 8 priorities, but no ToS information based on a 32 bit extension of Ethernet/Token Ring layer 2 header.
  - 802.1D includes 802.1p and 802.1Q (3 bit) extensions.

Value/Name	Examples	QoS Characteristics
7 Network Control	RIP, OSPF, BGP4	Critical
6 Voice	NetMeeting	Jitter-, latency-sensitive, low bandwidth
5 Video	PictureTel	Jitter-sensitive, high bandwidth
4 Controlled load	SNA transactions	Latency-sensitive
3 Excellent effort	SAP, SQL	Business critical, delay toleration
2 Best-effort	Best-effort	Best-effort
1 default	default	default
0 Background	Backups, FTP	Latency-insensitive

Classification of QoS Parameters



Network-dep., Communication-rel., Application-ori. QoS

- Network-dependent (N-QoS) are located in layer 1 and 2
  - N-QoS define features and characteristics of the underlying network, its topology, and the cabling.
  - N-QoS may be specified applying the network provider view concerning "Network Performance".
- Examples comprise:
  - ATM networks (ITU-T I.356): *E.g.*, Cell Error Ratio, Cell Loss Ratio, or Cell Transfer Delay.
- Communication-relevant (C-QoS) are located in layers 3 to 6
  - C-QoS define parameter for the commu-nication-relevant processing of user data
  - A subset of C-QoS determine performance behaviors, others define functionality.
  - Usage for end-to-end negotiations and resource reservation schemes is intended.
- Examples comprise:
  - Transport and Network layer protocols and
  - ATM protocols and X.25.
- Application-oriented (A-QoS) are located in layer 7
  - A-QoS define features and characteristics of the application
  - ATM protocols and X.25.
- Examples comprise:
  - Parameters for compression algorithms for audio/video, such as frame rate, burstiness, or frame size

Expressiveness and Architecture

- Performance-oriented QoS (P-QoS) defined by a name-value pair.
- A value itself is defined as quantifiable value, allowing for the content of numerical information.
- Functionality-oriented QoS (F-QoS) defined by a name-function pair.
- A function itself is defined as qualitative information, allowing for the comprehensive specification of a dedicated functionality.
- As a rule, the function comprises specifications of specific tasks or algorithms.
- Protocol-specific QoS defined within a communication protocol.
- A QoS parameter itself specifies protocol-internal information, such as window sizes, packet counter, or timer values.
- System-specific QoS defined within the communication subsystem, in particular the end-system.
- Such QoS parameters are dependent on the end-system and its architecture, *e.g.*, as for memory management schemes or scheduling schemes.

Type-of-Service (ToS) (1)

- There exist two basic Types-of-Service (ToS):
  - Best effort service and
  - Guaranteed service
- Best Effort Service:
  - Service type that does not deliver any guarantees of QoS parameter values (no commitment)
  - Reservation of resources within the end-system or the network has not taken place (no assurance)
  - Monitoring of QoS is not done, normally, due to the fact that no (re-)actions between service provider and user are defined
  - Specification of QoS parameters is not necessary
- Guaranteed Service:
  - Ensuring that QoS parameters are enforced in
    - Spatial or layered and
    - Temporal or session duration dimensions
  - Requires a reservation of resources

Type-of-Service (ToS) (2)

- Two different guarantees are possible:
  - Statistical (stochastic) guarantees – weak:
    - Utilization of network can be maximized (multiplexing).
    - Reserving resources for an "average" case necessary.
    - Temporary violations of guarantees possible.

$$P\,[Limit\,(QoS_p) \leq QoS_p\,(t)] \leq Plimit_p$$
$$P\,[Bound\,(QoS_p) \geq QoS_p\,(t)] \leq Pbound_p$$

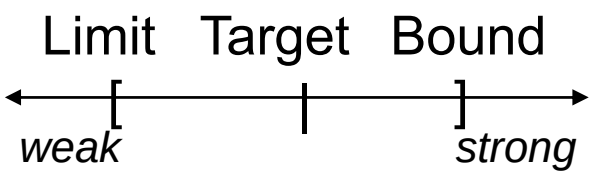
  - Deterministic guarantees – strong:
    - Overall peak-based approach (obligation of result).
    - Resource reservations are required for the worst case.

$$Limit\,(QoS_p) \leq QoS_p\,(t) \leq Bound\,(QoS_p)$$

- Prerequisites: QoS are specified, monitored, and enforced
  - ToS is sometimes called "QoS semantics" as well

ToS Specifications and Examples

- A Type-of-Service specification on a per QoS parameter basis is required, *e.g.*,
  - Bit error rate: guaranteed service (deterministic) and throughput: best effort
- In addition, 3 different values refine QoS semantics:
  - Limit: value that is weakest obligatory
  - Target: value that is expected to be central
  - Bound: value that is strongest obligatory

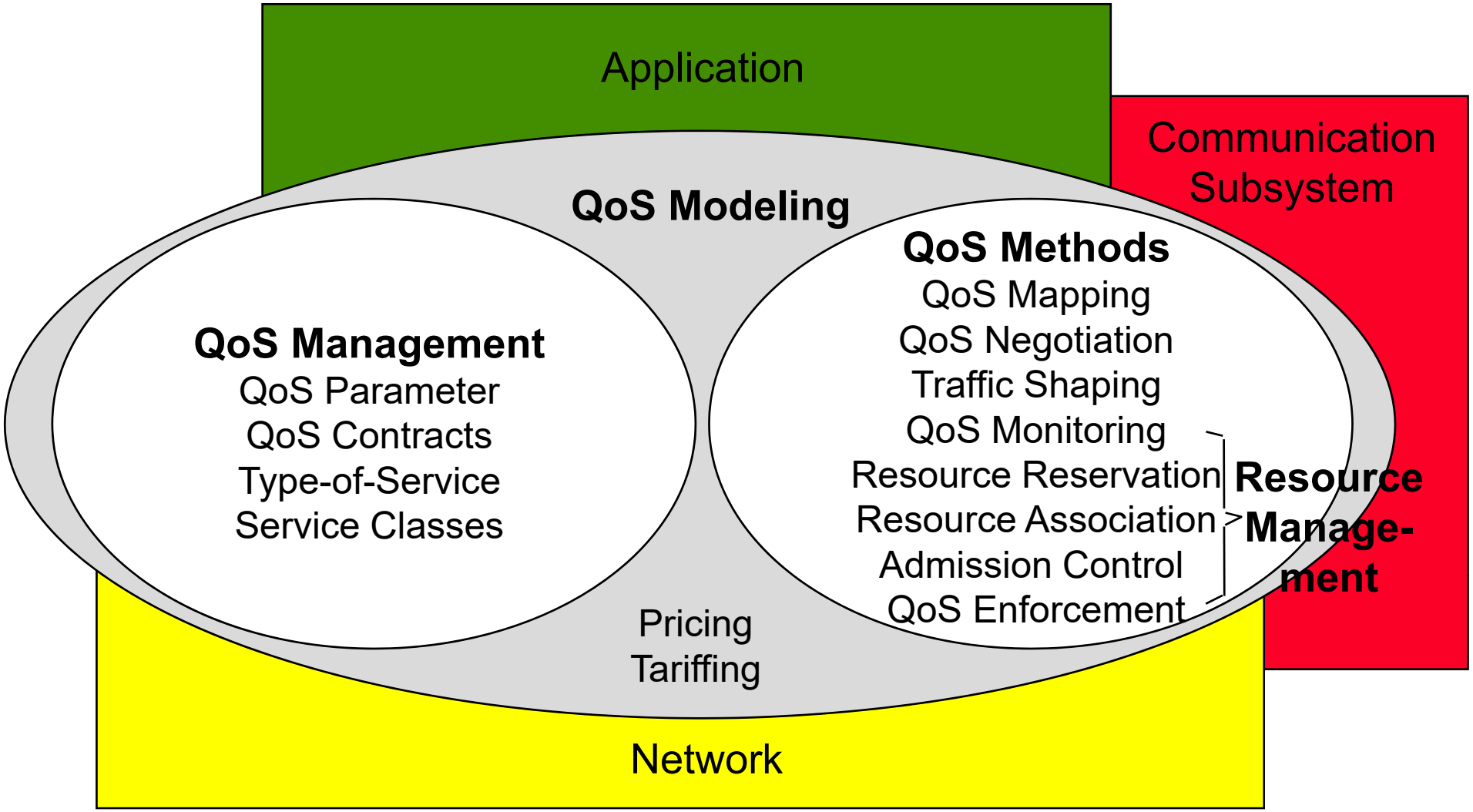


- File Transfer:
  - Best effort service concerning timing and delay:
    - No values can be specified or reserved
  - Guaranteed service (deterministic) concerning reliability:
    - Bit error rate is zero for received data (retransmission)
- Video Transmission:
  - Guaranteed service (statistic) concerning frame delay:
    - p percent of delayed frames may exceed the maximum bounded delay D
    - “Flickering” pictures (black outs) may occur due to late arriving frames

QoS Modeling

Generic QoS Model and Tasks
QoS Basic Framework
ITU-T G.1000
ATM QoS
Internet QoS
Comparison

Structuring QoS — QoS Model



Tasks and Components (1)

- QoS Parameter (QoS Syntax)
  - For specification purposes of application requirements, network prerequisites, or data flows a unique name-value pair is required.
- QoS Contract
  - Templates to specify the result of a QoS negotiation and the final resource reservation and admission.
  - They contain certain agreed upon QoS parameter values and traffic patterns for a particular data flow or entire application.
  - Flow specifications include (parts of) QoS contracts.
- Type-of-Service (QoS Semantics)
  - Definition of semantics on a per QoS parameter basis for a special service.
- Service Classes
- QoS Methods
  - QoS Mapping (QoS Transformation)
    - Mapping QoS of different views. Traffic Shaping
    - Shaping user data to be transmitted, to comply with the agreed upon QoS contract, in particular the traffic shape.

Tasks and Components (2)

- QoS Negotiation
  - Methods to adjust or align distinct QoS of different peers to a commonly accepted end-to-end QoS.
  - Methods to re-adjust and re-align QoS for an already negotiated QoS are included and called QoS Re-negotiation.
- QoS Monitoring
  - Measurements and supervision of current values and behavior of negotiated and contracted QoS parameter.

Resource Management (1)

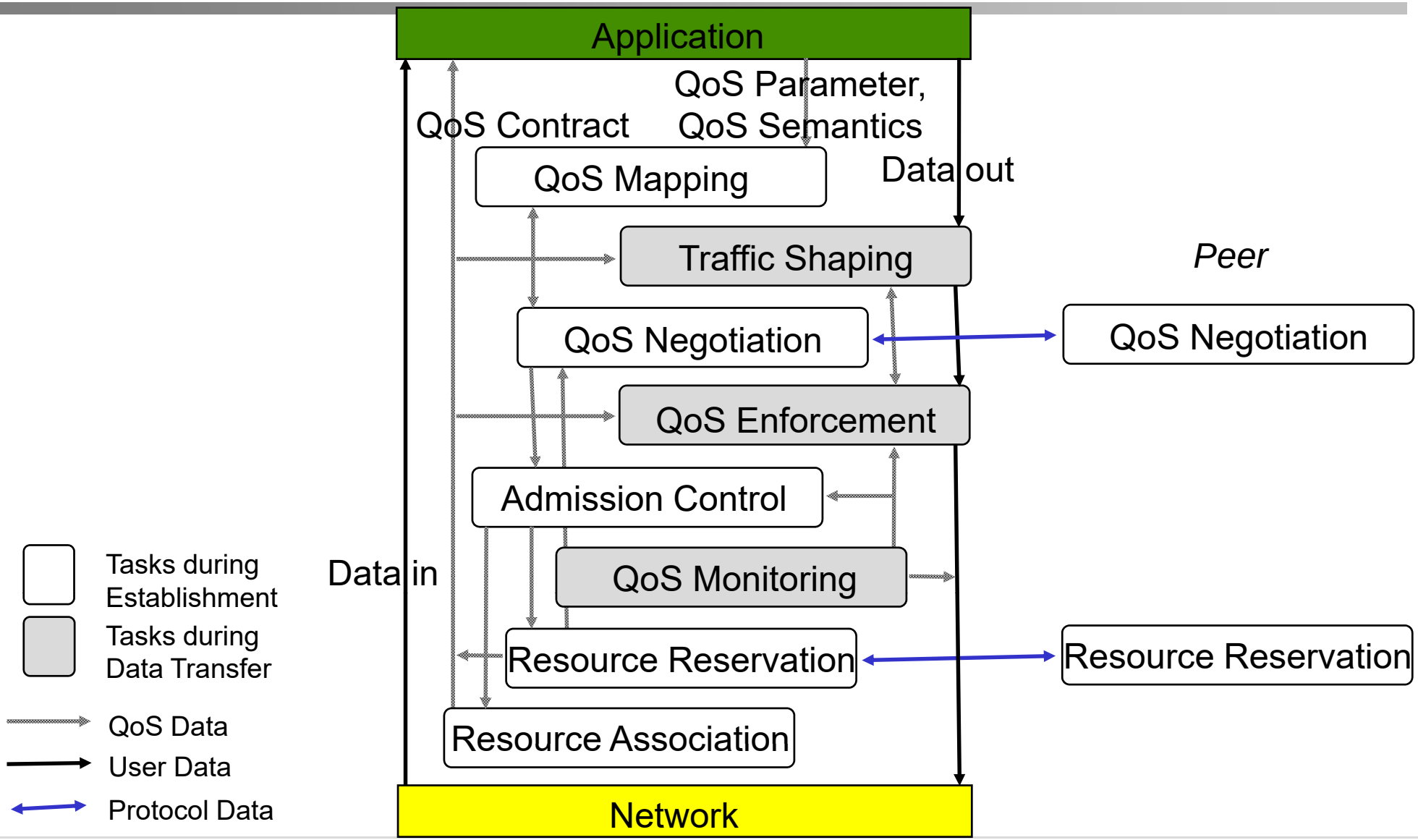
- Resource Reservation
  - Protocols to reserve resources within networks on a neighbor system basis (link-by-link) and peer basis (end-to-end) to provide guaranteed services within the network.
- Resource Association
  - Mechanisms to associate resources within an end-system based on a given a traffic shape and on the computed QoS of this resource provided by the admission control.
- Admission Control
  - Mechanisms to allow for accepting or rejecting new or joining flows for a given architecture including specified resources.
  - Included are mechanisms to determine performance of a given resource under current traffic conditions (QoS computation).

Tasks and Components (3)

- Resource Management (2)
- QoS Enforcement
  - Based on monitored QoS, mechanisms to enforce the QoS contract and to maintain current QoS as close as possible, while controlling and scheduling end-system resources.
- Pricing and Tariffing
  - Determination of a scheme that maps delivered services onto monetary units (prices) in an open market or rates in a regulated market (tariffs).

- Tasks: informal functional description of a QoS model
- Components: abstract architectural units for performing one or multiple of these tasks
  - QoS models define all required components that include exactly one single task, each to be performed separately (modularization principle)
  - Definite design (concrete implementation) may combine tasks to a component to:
    - Optimize storage/access of frequently used QoS data
    - Minimize processing delays internally in the end-system
    - Simplify inter-component communications

Generic QoS Model

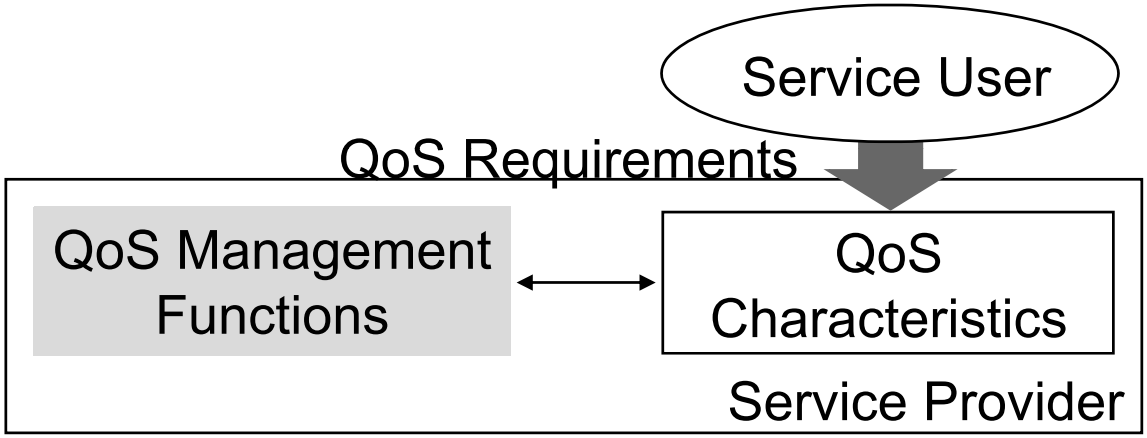


ISO QoS Framework (QFw)

- QoS Framework (ISO N9680 1995, ITU X.641 1997):
  - Concepts and relationships between QoS
  - Guidance on QoS applicable to communication systems
  - Mechanisms to manage, negotiate, and meet QoS
- Main definitions:
  - Quality-of-Service (QoS):
    A set of qualities related to the collective behavior of one or more objects
  - QoS characteristic:
    A quantifiable aspect of QoS, which is defined independently of the means by which it is represented and controlled
- Characteristics:
  - QoS Framework covers a wide range of tasks
  - Basic classification of tasks for QoS management has been done
  - Terminology is defined provisionally (never updated since then)
  - Many details as being required for a complete QoS model are missing, such as resource association, admission control, or QoS maintaining mechanisms

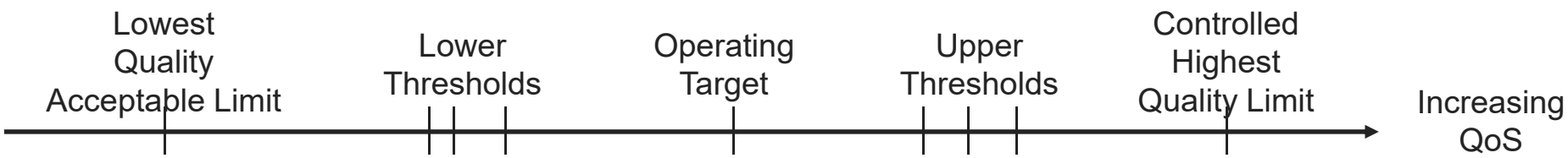
QFw – QoS Concept

- The concept contains descriptions of QoS concerning:
  - QoS requirements and policies and QoS management functions implementing them,
  - QoS categories that represent the particular set of user requirements for QoS,
  - QoS characteristics that describe the fundamental aspects of QoS and that are to be managed, and
  - Basic QoS mechanisms that are combined to form QoS management functions.

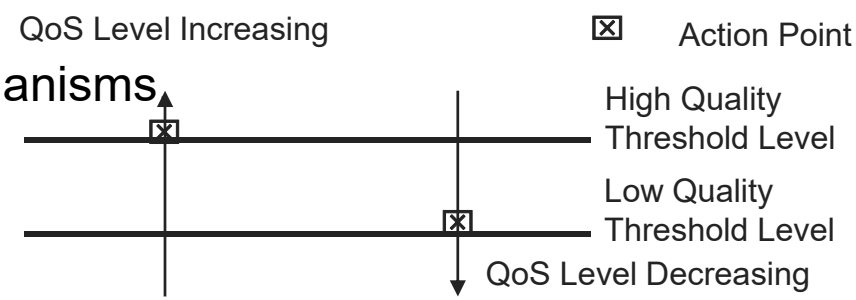


QFw – QoS Requirement Parameter Values (1)

- QoS semantics
  - Values of parameters that QoS requirements may take
  - Limits and thresholds associated with actions are referred to as “trigger points”
  - Particular QoS requirement may include one or more of such values
  - Abstract “low” and “high” qualities used instead of numerical values



- QoS operating targets are negotiated, while
  - Lowest quality acceptable limit should not be fallen below
  - Controlled highest quality limit should not be gone above
  - The strength of prosecution crossing a limit determined by negotiated QoS mechanisms
  - Thresholds define “action points” no prosecution is started, crossing allowed

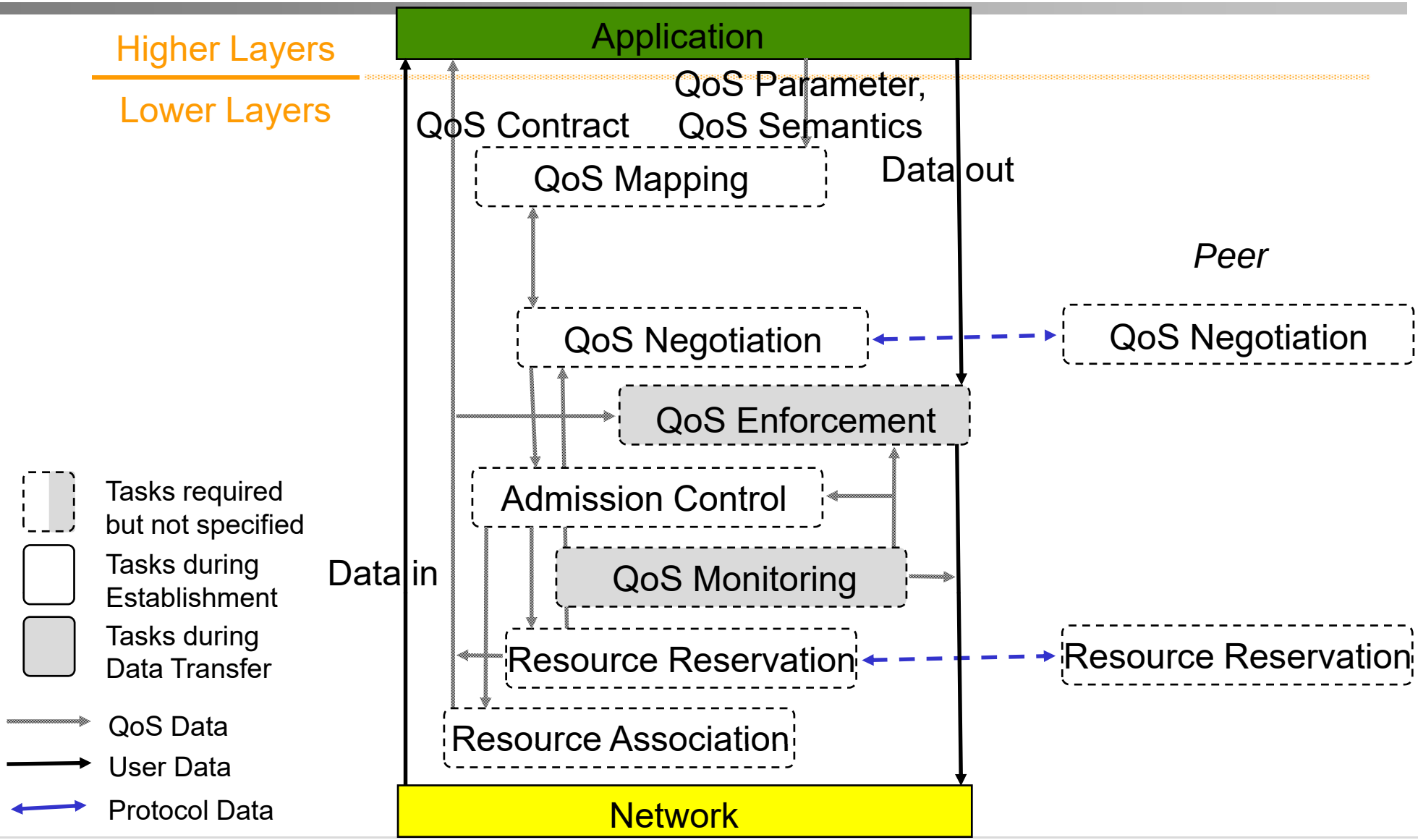


QFw – QoS Requirement Parameter Values (2)

- Actions to be taken encompass:
  - Take no action,
  - Modify the operation of the service provider to keep within the limits negotiated,
  - Store a value for future reference (*e.g.*, for enquiry purpose),
  - Send a signal (*e.g.*, alert or alarm), or
  - Abort the service.

- Levels of agreement include:
  - Best effort (weak and no assurance),
  - Compulsory (medium, monitoring, degradation, abortion), or
  - Guaranteed (initiation after full maintenance is guaranteed, baring “rare” failures, such as system crashes).

Mapping QFw onto the Generic QoS Model



ITU-T G.1000 (1)

- Communications Quality-of-Service: A Framework and Definitions (Draft started: Fall 2001, Final 2003)

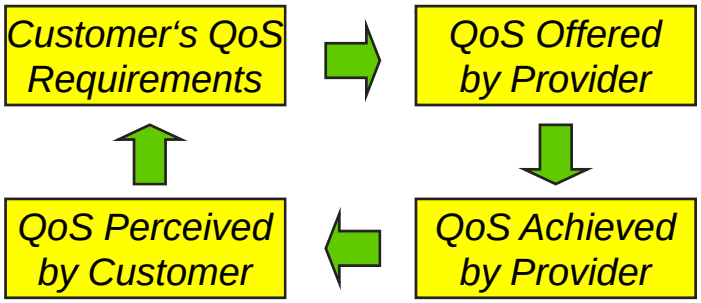
- Scope:
  - Consistent and uniform approach to QoS, IP areas
  - Set of well-defined, relevant, customer-effecting approach
  - To plan and to deploy networks
  - To monitor service quality

- References:
  - E.800: QoS and Network Performance
  - G.1010: End-user Multimedia QoS categories
  - I.350: General Aspects of QoS, including ISDN

ITU-T G.1000 (2)

- QoS requirements of user/customer:
  - Statement of level of quality required by applications of customers/users of a service (non-technically)
- QoS offered/planned by a provider:
  - Statement of level of quality expected to be offered to customer by service provider
- QoS delivered/achieved by a provider:
  - Statement of level of the actual quality achieved and delivered to the customer
- QoS perceived by user/customer:
  - Statement expressing level of quality that customers believe they have experienced

- Four view points of QoS:
  - Customer‘s QoS requirements (-> G.1010)
  - Service provider offerings of QoS (-> SLAs)
  - QoS achieved or delivered (-> SLAs)
  - Customer survey ratings of QoS



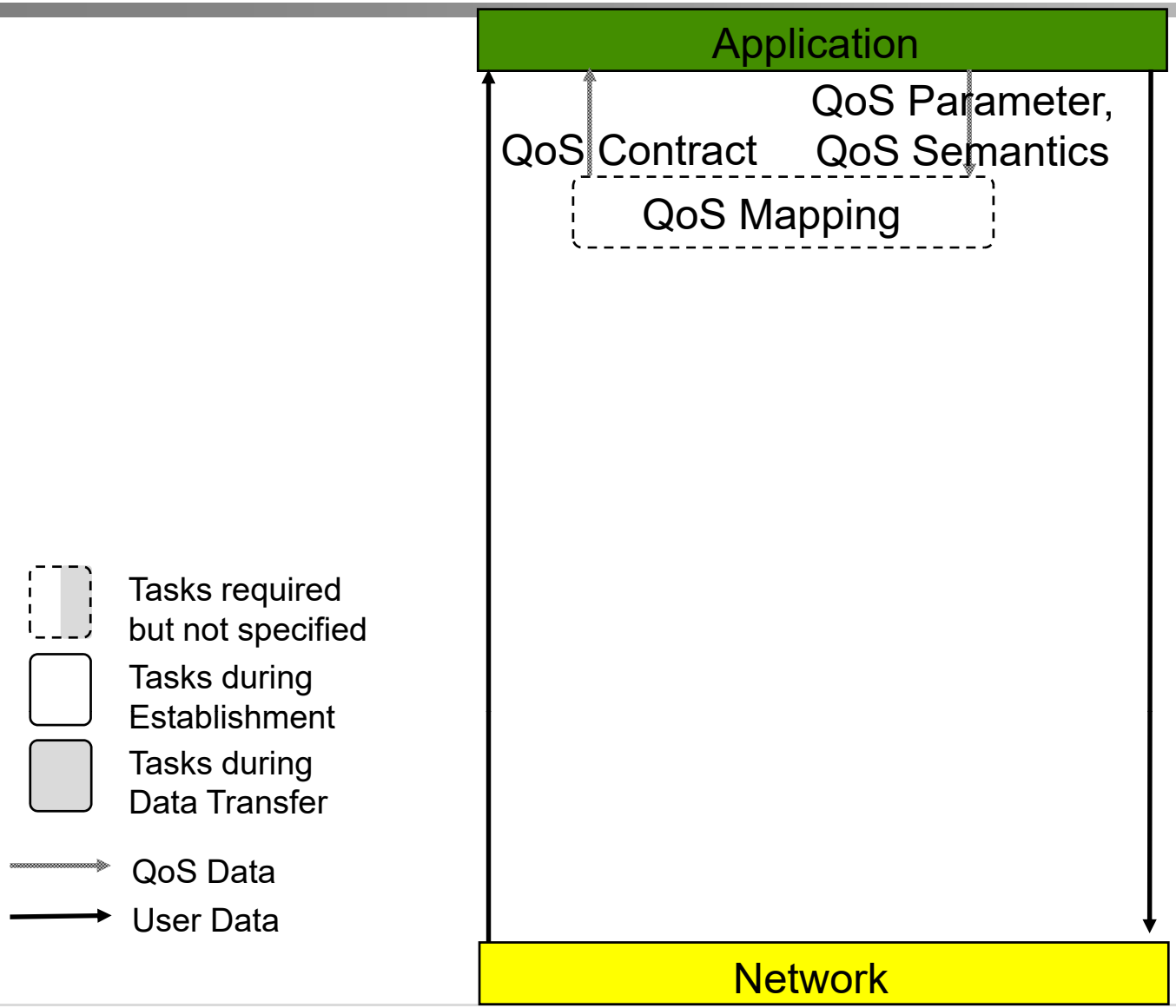
ITU-T G.1000 (3)

Service Quality Criteria

Service Function	Speed	Accuracy	Availability	Reliability	Security	Simplicity	Flexibility
SERVICE MANAGEMENT:							
Sales and Pre-contract Activities							
Provision							
Alteration							
Service Support							
Repair							
Cessation							
CONNECTION QUALITY:							
Connection Establishment							
Information Transfer							
Connection Release							
BILLING							
NETWORK/SERVICE MANAGEMENT BY CUSTOMER							

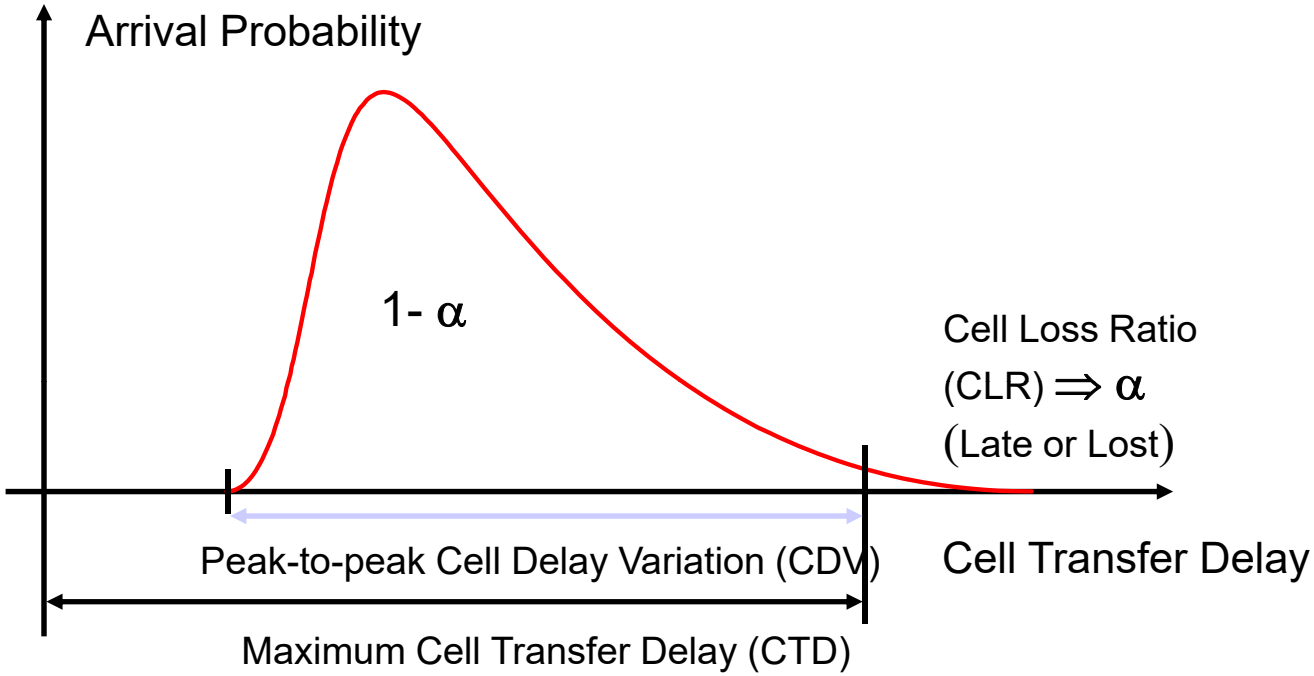
Matrix for the determination of communication QoS

Mapping G.1000 onto the Generic QoS Model



ATM QoS and Model for Cell Transfer Times

- QoS in ATM deals with multiplexing many flows on link
  - May lead to a correlation between flows
  - Aggregated view of all flows in total lacks significant information for guarantees
- ATM bearer services supplied with dedicated QoS on 3 levels
  - Call level,
  - Connection level, and
  - Cell level.
- ATM Traffic Management:
  - ITU-T and ATM Forum approaches.
- ATM model for cell transfer times:



ITU–T I.362 Service Classes for AAL

QoS	Class A	Class B	Class C	Class D
Timing between source and sink	required		not required	
Bit rate	constant	variable		
Connection	connection-oriented			connectionless
Examples	Circuit-emulation (fixed rate V)	Variable rate (A/V)	Traditional connection-oriented data	Intra-LAN traffic (data)

A: Audio; AAL: ATM Adaptation Layer; LAN: Local Area Network; V: Video

ATM Forum Traffic and QoS Parameters

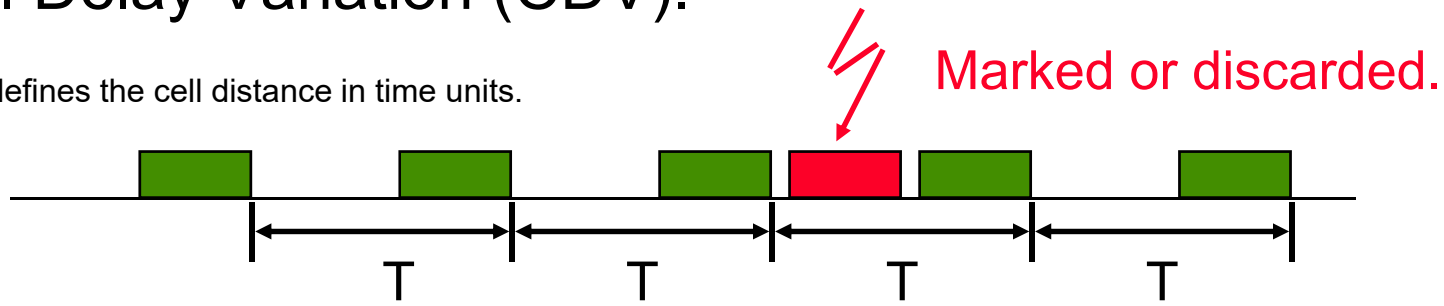
- Traffic parameters define the sending behavior of end-systems which are valid per connection:
  - Peak Cell Rate (PCR),
  - Sustainable Cell Rate (SCR),
  - Maximum Burst Size (MBS),
  - Minimum Cell Rate (MCR), and
  - Cell Delay Variation Tolerance (CDVT).

- Negotiated QoS parameters, quality of a connection:
  - Maximum Cell Transfer Delay (CTD),
  - Peak-to-peak Cell Delay Variance (CDV), and
  - Cell Loss Ratio (CLR).

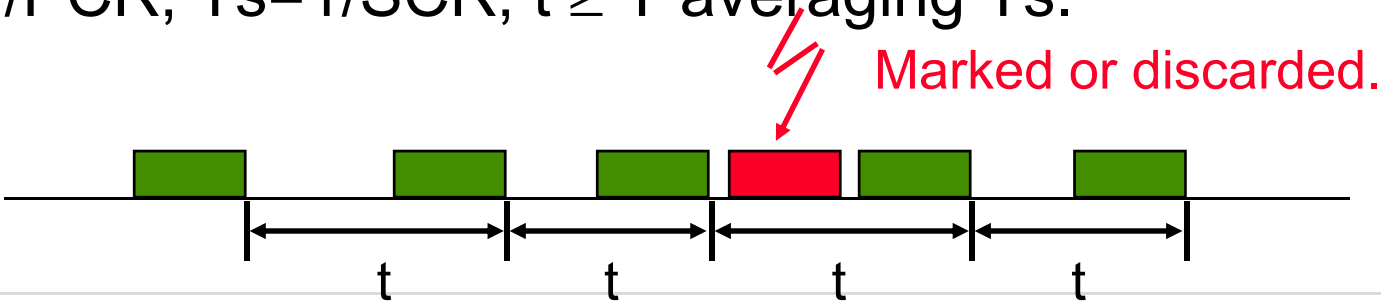
ATM Forum – Real-time Services

- CBR: Traffic with constant Peak Cell Rate (PCR) or the period T=1/PCR, maximum Cell Transfer Delay (CTD), and Cell Delay Variation (CDV).

PCR defines the cell distance in time units.



- rt-VBR: Traffic with PCR, Sustainable Cell Rate (SCR), Maximum Burst Size (MBS), maximum CTD, and CDV. T=1/PCR, Ts=1/SCR, t ≥ T averaging Ts.



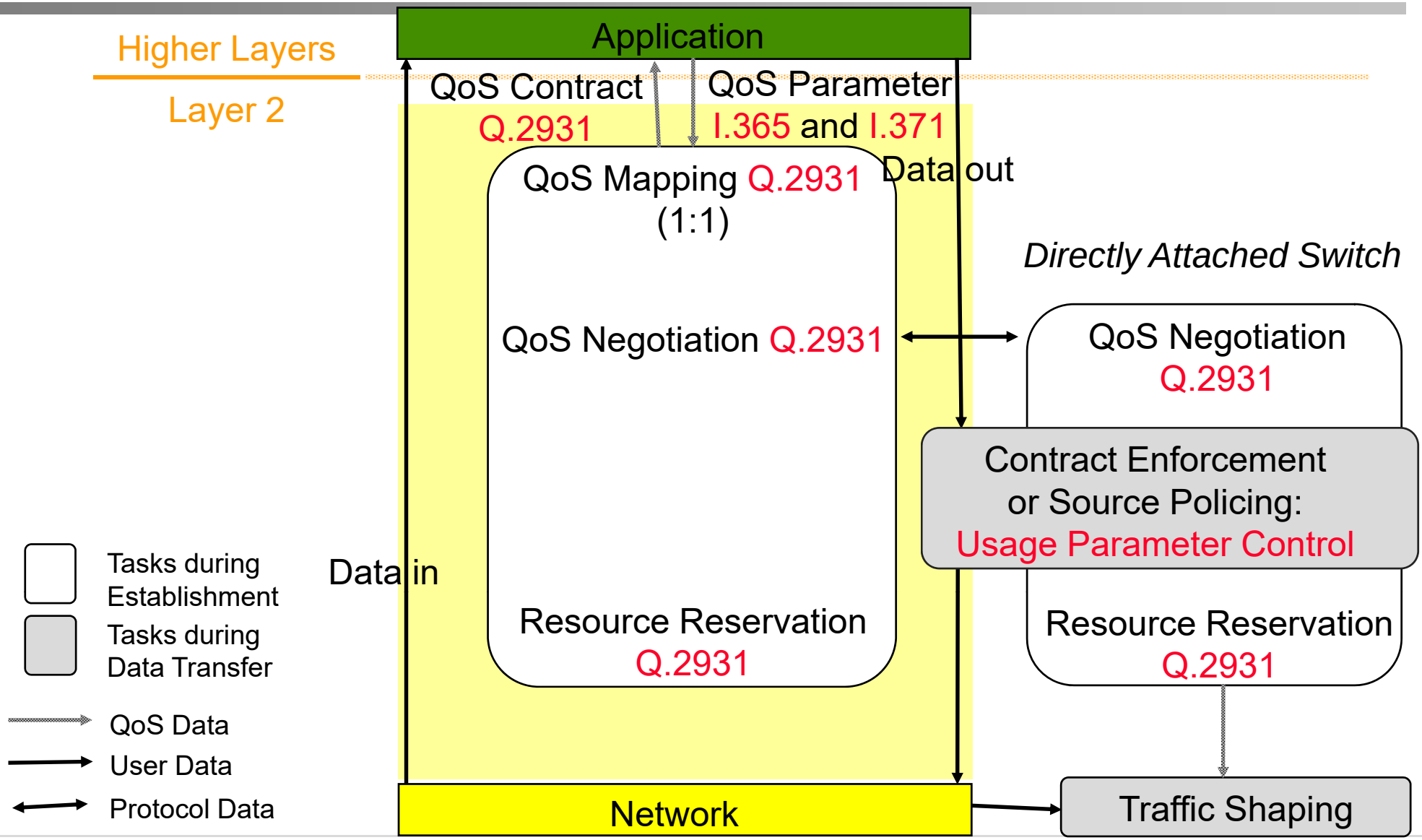
ATM Layer Service Categories

QoS	CBR	rt-VBR	nrt-VBR	UBR	ABR
PCR and CDVT	specified	specified	specified	specified*	specified
SCR, MBS, CDVT	N/A	specified	specified	N/A	N/A
MCR	N/A	N/A	N/A	N/A	specified
Peak-to-peak CDV	specified	specified	unspecified	unspecified	unspecified
Max. CTD	specified	specified	unspecified	unspecified	unspecified
CLR	specified	specified	specified	unspecified	low
Feedback	unspecified	unspecified	unspecified	unspecified	specified

PCR: Peak Cell Rate
CDVT: Cell Delay Variation Tolerance
SCR: Sustainable Cell Rate
MBS: Maximum Burst Size
MCR: Mean Cell Rate
CDV: Cell Delay Variation
CTD: Cell Transfer Delay

CLR: Cell Loss Ratio
QoS: Quality-of-Service Parameter
CBR: Constant Bit Rate
VBR: Variable Bit Rate
UBR: Unspecified Bit Rate
ABR: Available Bit Rate
(n)rt: (non) real-time
*: informational

Mapping ATM onto the Generic QoS Model



QoS in IP

- Quality-of-Service in IP (Internet Protocol) is defined in RFC 791 as a bit-vector (so-called Type-of-Service):
  - 4 priority levels (2 bit),
  - Delay: normal and low (1 bit),
  - Throughput: normal and high (1 bit), and
  - Reliability: normal and high (1 bit).

Priority	Delay	Throughput	Reliability	0	0

- Usage of these information is optional.
- Many router and switching elements do not process the Type-of-Service vector:
  - IP does not offer any service guarantees at all.

QoS in IPv6

- Concept of flows:
  - A packet sequence is considered as a flow, while each packet is identified by the source address and a flow label.
    - Flow labels are random numbers and used to access a flow state via a hash-table.
  - Non-default or real-time services are explicitly specified:
    - Set-up,
    - In-packet options, or
    - Other means.

- Priorities are used to identify packets relative to other packets of the same source.
  - 0 – 7: congestion-sensitive traffic; 8 – 15: drop-priority.

Differentiated Services

- Definition of externally observable forwarding behaviors of routers: Per-hop Behavior (PHB).
  - Expedited Forwarding Service.
  - Assured Forwarding Service.

- Re-use of Type-of-Service field of the IP packet:

Precedence Field	Type of Service Field	0
3 bit	4 bit	

as Differentiated Services Field for Code Points (CP):

← DSCP →		
Class Selector CP		CU
3 bit	3 bit	

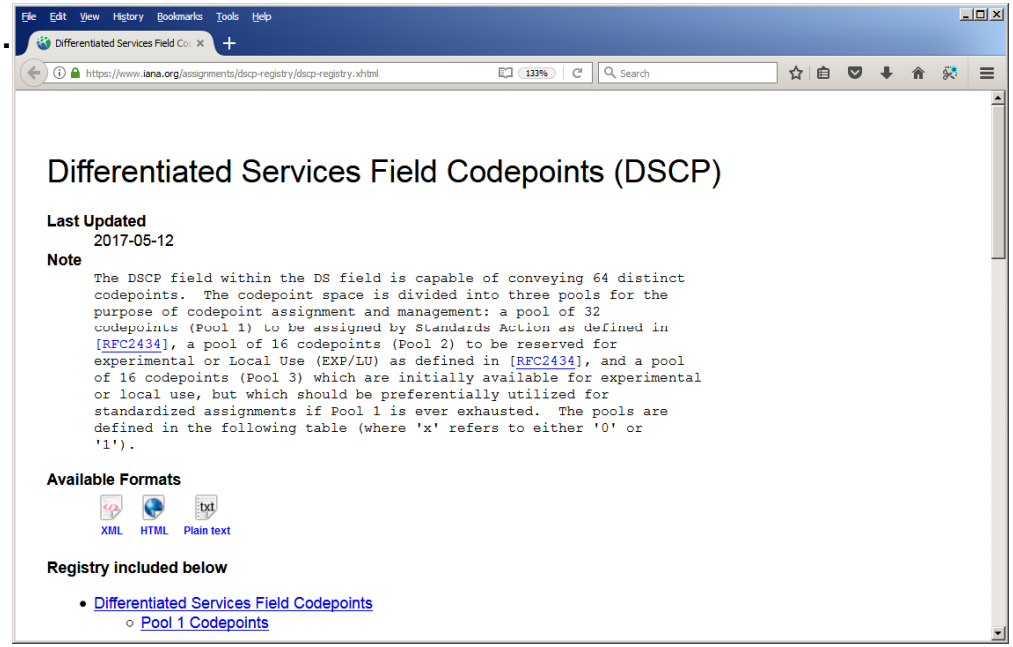
CU: Currently Unused

Differentiated Services Code Points

- DS field with no backward compatibility to ToS.
- Three code point pools:
  - Standard and two types of local use or experiments.

Pool	CP	Policy
1	xxxxx0	Standard
2	xxxx11	Local type 1
3	xxxx01	Local type 2

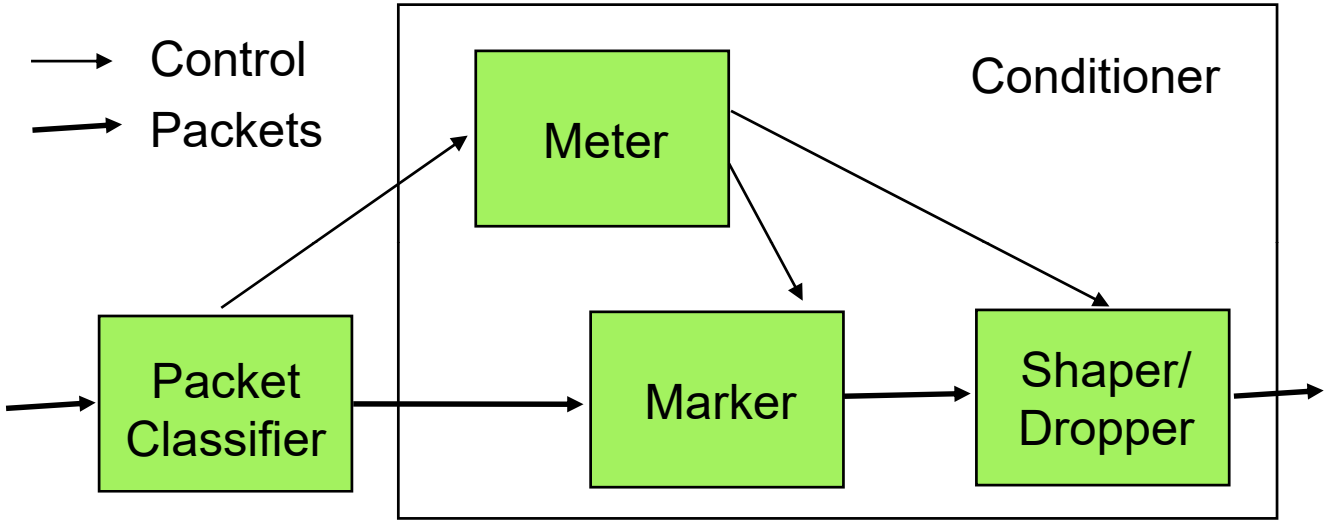
- Code point grouping:
  - 101000, 101001, 101010, 101011: 1010xx similarly treated by a router.
  - Each group mapped onto a PHB value.



- PHB value mapped to a DSCP value being part of the DiffServ field.
  - Mapping not static.
  - Option: Several PHBs to be mapped into a single DSCP.

DiffServ Implementation

- Differentiated Services Architecture components:
  - Classifier and conditioner.
  - Conditioner must not contain all components:
    - Egress, ingress, or first-hop router different demands.
    - Service-dependent.



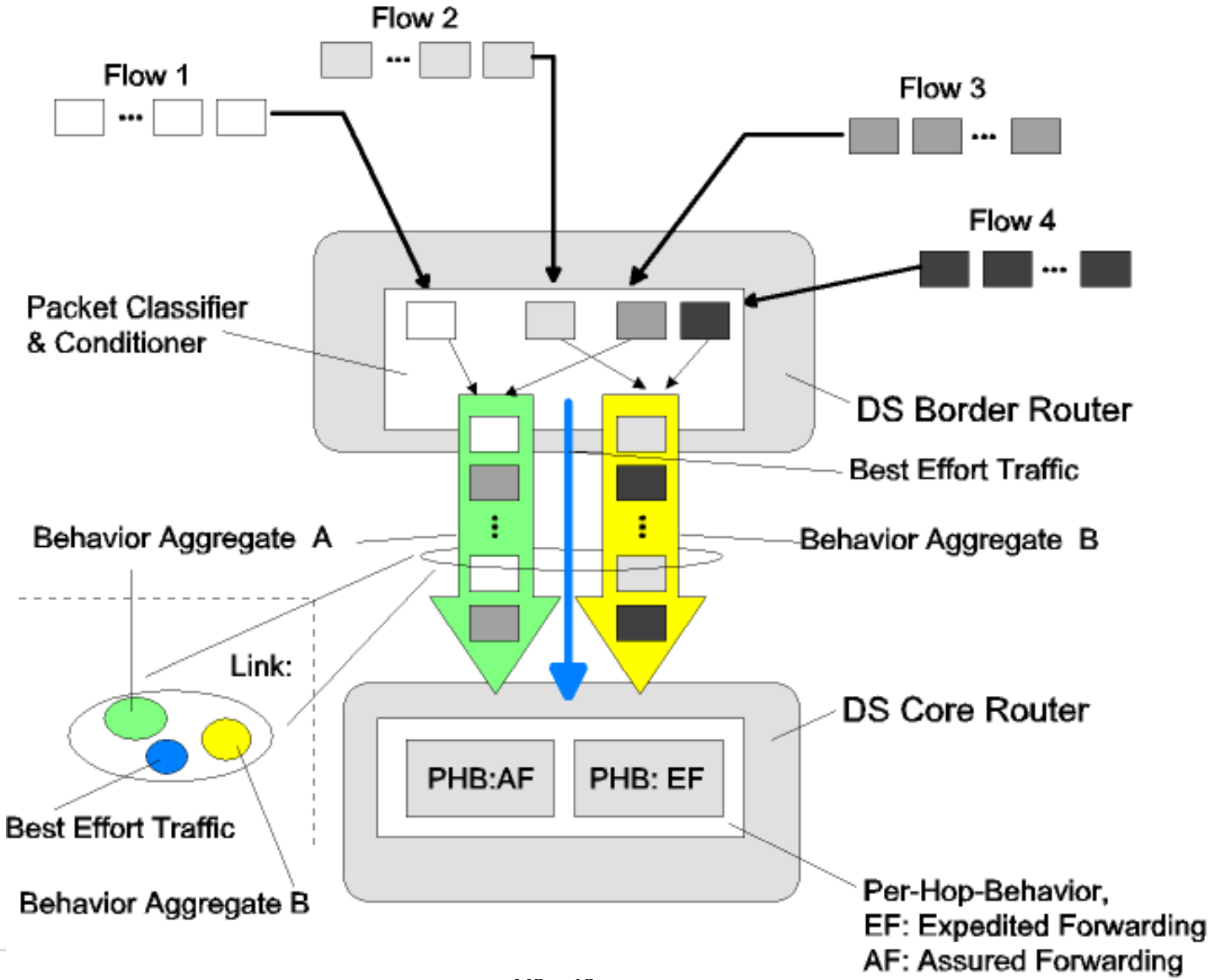
DiffServ Components

- Packet classifier:
  - Behavior Aggregate (BA) classifier selects packets based on the DSCP only.
  - Multi Field (MF) classifier selects packets based on customer-specific header fields.
- Traffic profile:
  - Determining whether a packet is in-profile or out-of-profile.
  - Specified in a Traffic Conditioning Agreement (TCA).
  - *E.g.*, token bucket:

```
If Code_point = X then use token_bucket(r, b)
    if token available then forward packet
    else discard packet
```

- Traffic conditioner:
  - Responsible to set DSCP of packet according to classification result
  - Traffic meter: Measurement of temporal stream properties
  - Packet marker: Set the DS field of a packet, adding the packet to a particular BA
  - Shaper: Delaying one or all packets of stream to be compliant to a traffic profile
    - Discarding may happen, due to limited buffer size.
  - Dropper: Discard one or all packets of stream to be compliant to a traffic profile (policing). Droppers may be implemented as a special case shaper (buffer size = 0).

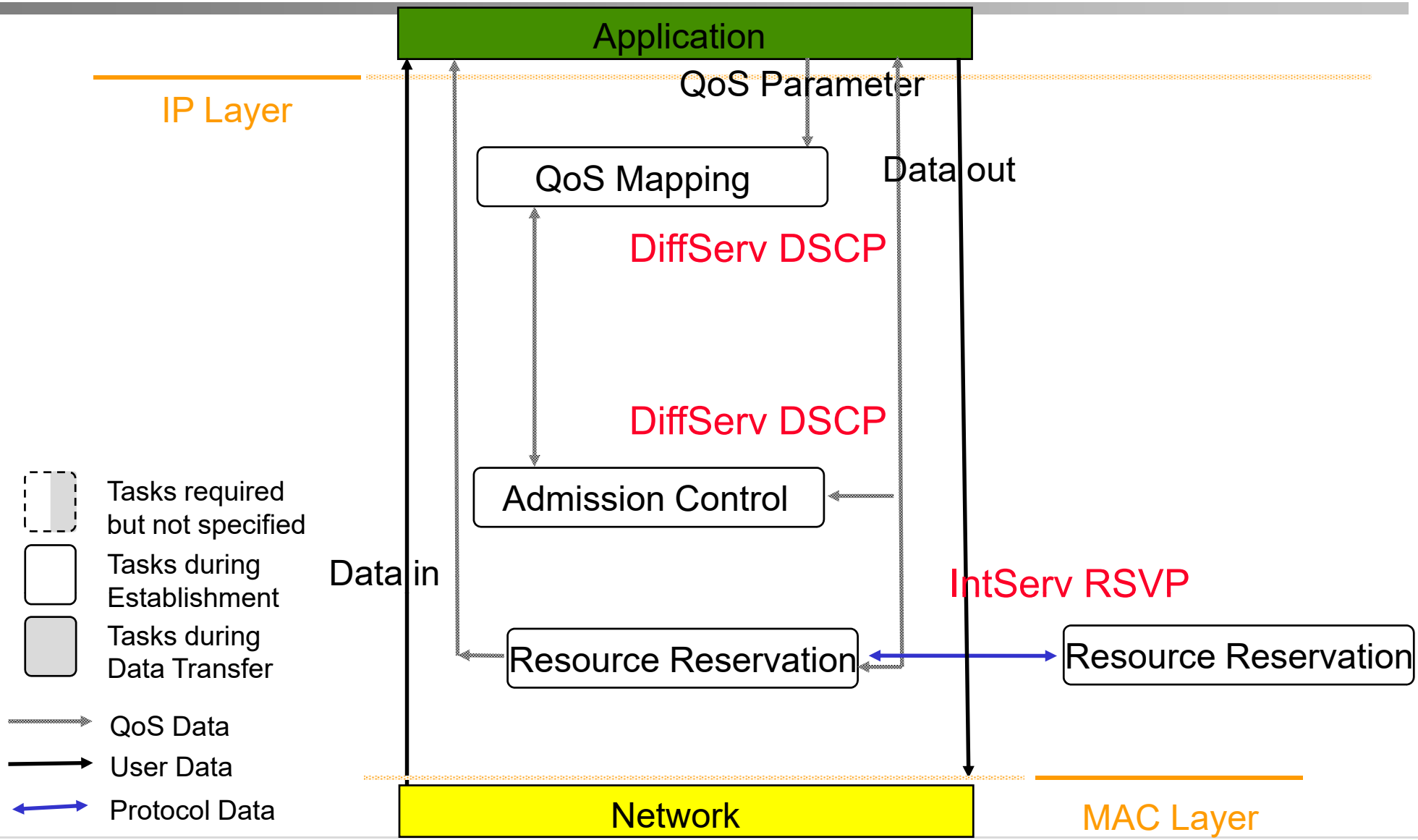
DiffServ Summary



DiffServ Problems

- Problems with expedited and assured forwarding:
  - Edge-based admission control does not avoid congestion within this upper service class, if upper class users and their traffic profiles are not known.
  - Choosing a correct user profile (for an organization) and its parameter setting for aggregated traffic is non-trivial.
  - Selection of particular user profiles via configuration, signaling protocols, or management protocols requires effort.
  - Traffic directions: from vs. to the ISP. Sender profiles may be known, however, receiver profiles remain unknown.
  - Starvation of lower class traffic, that is best-effort traffic, is possible.
  - Limited protection against misbehaving flows.

Mapping IntServ/DiffServ onto the Generic QoS Model



Comparison of Generic QoS Model Mappings

Criteria	ATM	QFw	OSI´95	F-CSS	QoS-A	Broker
QoS Management:						
QoS Parameter	*defined*	*defined*	*defined*	*defined*	*defined*	*defined*
QoS Semantics	*defined*	*defined*	*defined*	*defined*	*defined*	*mentioned*
Traffic Classes	*defined*	*no*	*no*	*yes*	*mentioned*	*mentioned*
QoS Contract	*defined*	*mentioned*	*mentioned*	*yes*	*yes*	*yes*
Flow Specification	–	–	–	*yes*	*yes*	*yes*
QoS Methods:						
QoS Mapping	*no*	*mentioned*	*mentioned*	*mentioned*	*yes*	*yes*
Traffic Shaping	*yes*	*no*	*no*	*no*	*yes*	*yes*
QoS Negotiation	*yes*	*yes*	*yes*	*yes*	*yes*	*yes*
QoS Enforcement	*no*	*mentioned*	*no*	*mentioned*	*mentioned*	*yes*
Admission Control	*yes*	*mentioned*	*no*	*no*	*no*	*yes*
QoS Monitoring	*no*	*mentioned*	*mentioned*	*mentioned*	*mentioned*	*mentioned*
Resource Reservation	*yes*	*mentioned*	*no*	*mentioned*	*mentioned*	*mentioned*
Resource Association	–	*mentioned*	*no*	*mentioned*	*mentioned*	*mentioned*

mentioned: Necessity mentioned
yes: QoS method is explicitly applied
defined: QoS management is defined
no: QoS method is not supported

QoS Methods

Traffic Engineering and Models
QoS Mapping
QoS Negotiation
Admission Control
Traffic Shaping
QoS Monitoring
QoS Enforcement
QoS Contracts

Traffic Engineering and Policies

- Traffic Engineering
  - Task of mapping traffic flows onto an existing physical topology.
  - Goal 1: Minimization of packet loss and packet delay
  - Goal 2: Optimization of network resources (avoiding overload situations, balancing)
- Traffic engineering “applications” allow for a precise control of how traffic flows are placed within a routing domain. Consists of:
  - Traffic management (short-term) and
  - Network planning (long-term)
- Traffic management
  - Set of policies/mechanisms for satisfying range of different application service requests
  - Acting across: diversity and efficiency.
  - Subsumes traditional ideas of congestion control:
    - Overloaded resource suffers from service degradation.
    - Policies scale back demand or restrict access.

Traffic Models

- Goal of effectively managing traffic requires:
  - Requirements of individual applications and organizations.
  - Their typical „behavior“.

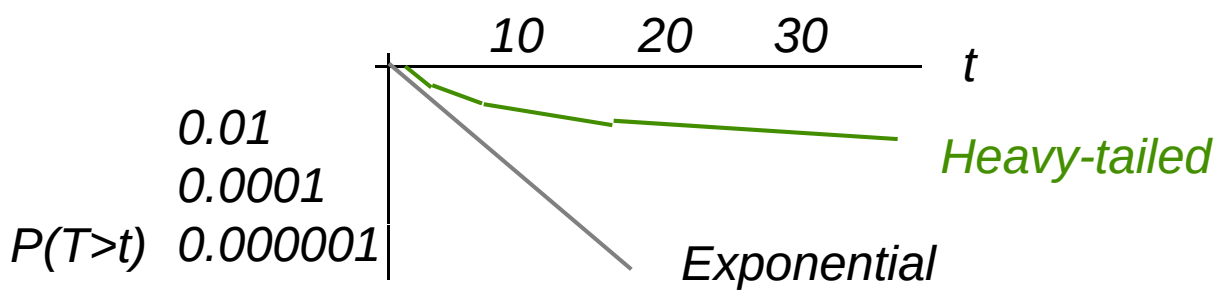
- Traffic Models:
  - Summarize „expected behavior“.
  - Obtained by detailed traffic measurements or amenable to mathematical analysis.
  - State of the art in traffic modeling:
    - Telephone traffic model
    - Internet traffic model

  - Change of applications make these models to change!

Telephone Traffic Model

- Call arrival model:
  - How are calls placed?
  - Inter-arrival times drawn from an exponential distribution (poisson process models all arrivals).
  - Memory-less (certain time elapse does not tell the future).

- Call holding-time model:
  - Call holding-times drawn from an exponential distribution:
    - Call longer than x decreases exponentially with x.
  - Heavy-tailed distribution in “recent” studies (1990s):

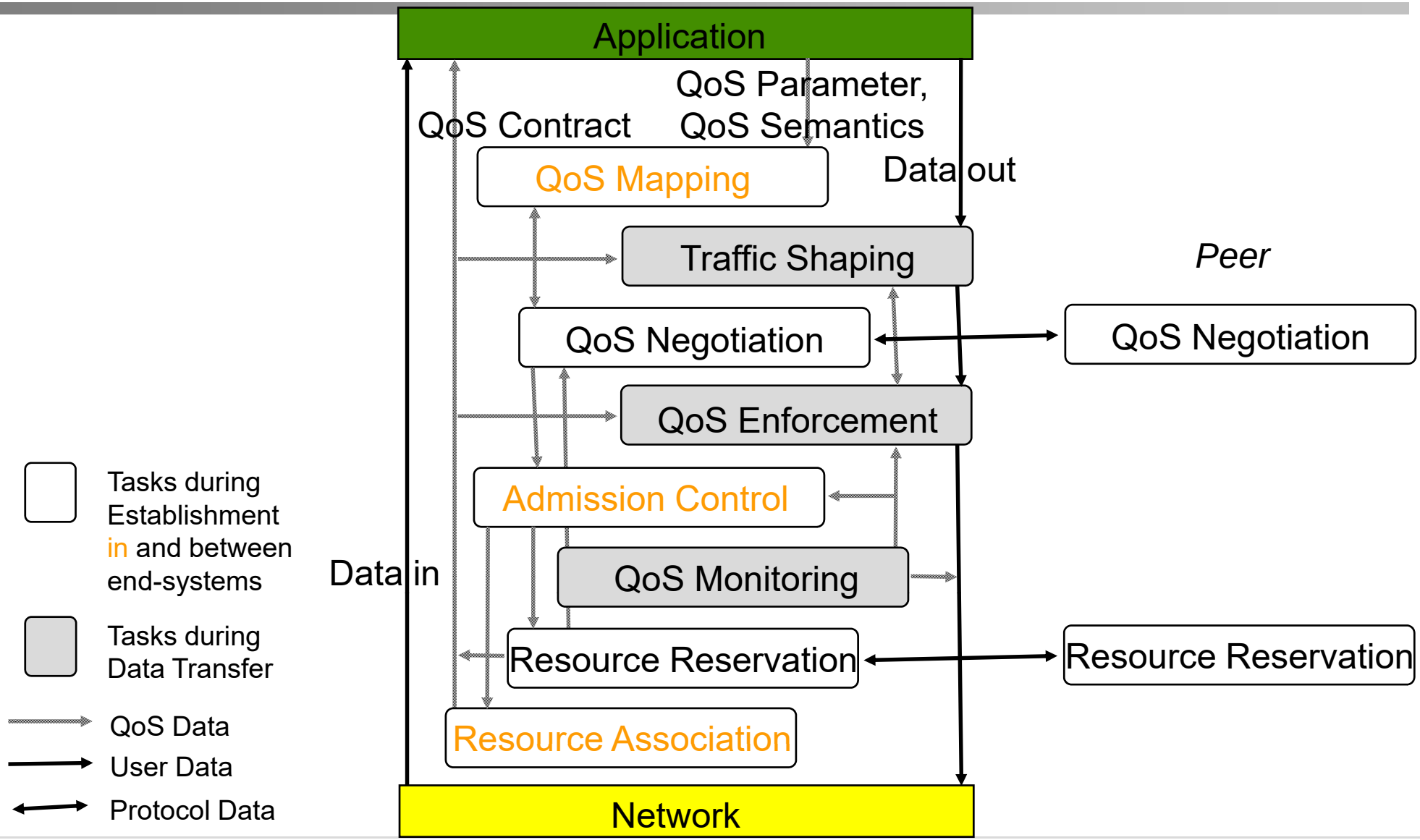


Internet Traffic Model

- Parameters to characterize applications:
  - Distributions of inter-arrival times between app. invocations.
  - Duration of a connection.
  - Number of bytes transferred during a connection.
  - Inter-arrival times of packets within a connection.

- Note: There is little consensus on models!
  - *E.g.*, inter-arrival times: Exponential or Weibull.
  - Effective means: Measurements to fit to statistical model.
  - LAN traffic differs heavily from WAN traffic.
    - More local bandwidth, tendency for longer holding times, higher peak data rates.
    - Expensive wide area bandwidth, less volume.

Generic QoS Model – Focus on QoS Methods



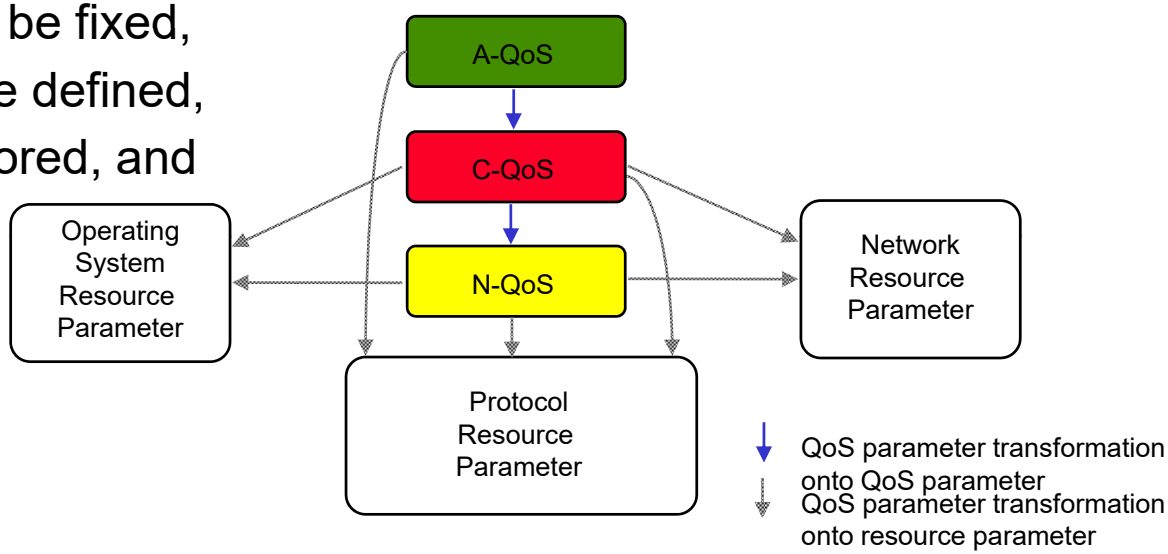
Time-scales of QoS Methods (Management)

Time Scale	Mechanism	Net	End-system
Less than one RTT (Cell level)	Scheduling, buffer management	x	x
	Regulation, policing	x	x
	Routing (connection less)	x	x
	Error detection and correction	x	x
One or more RTTs (Burst level)	Feedback flow-control	x	x
	Retransmission		x
	Renegotiation	x	x
Session (Call level)	Signaling	x	x
	Admission control	x	
	Service or congestion pricing	x	
	Routing (connection-oriented)	x	
Day	Peak-load pricing	x	
Weeks and more	Capacity planning	x	

QoS Mapping

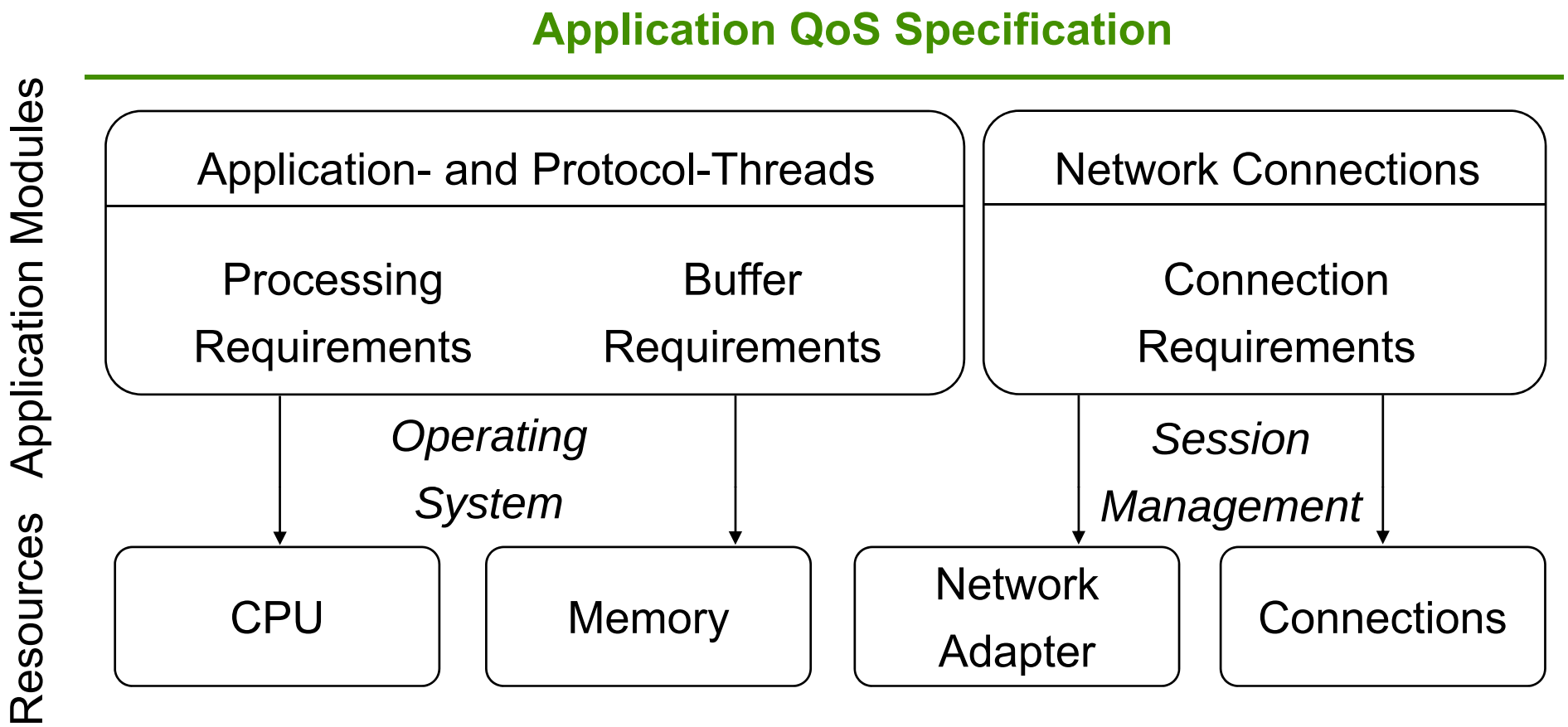
- Method QoS mapping defines translation procedures for QoS
- Targets:
  - Provides a comparable view of requirements,
  - Offers the potential to negotiate QoS, and
  - Allows for processing admission control tests

- Assumptions:
  - Considered QoS parameters to be fixed,
  - Available limits/boundaries to be defined,
  - Field of applications to be anchored, and
  - Operating system, network, and protocol resource parameters are specialized QoS parameters.



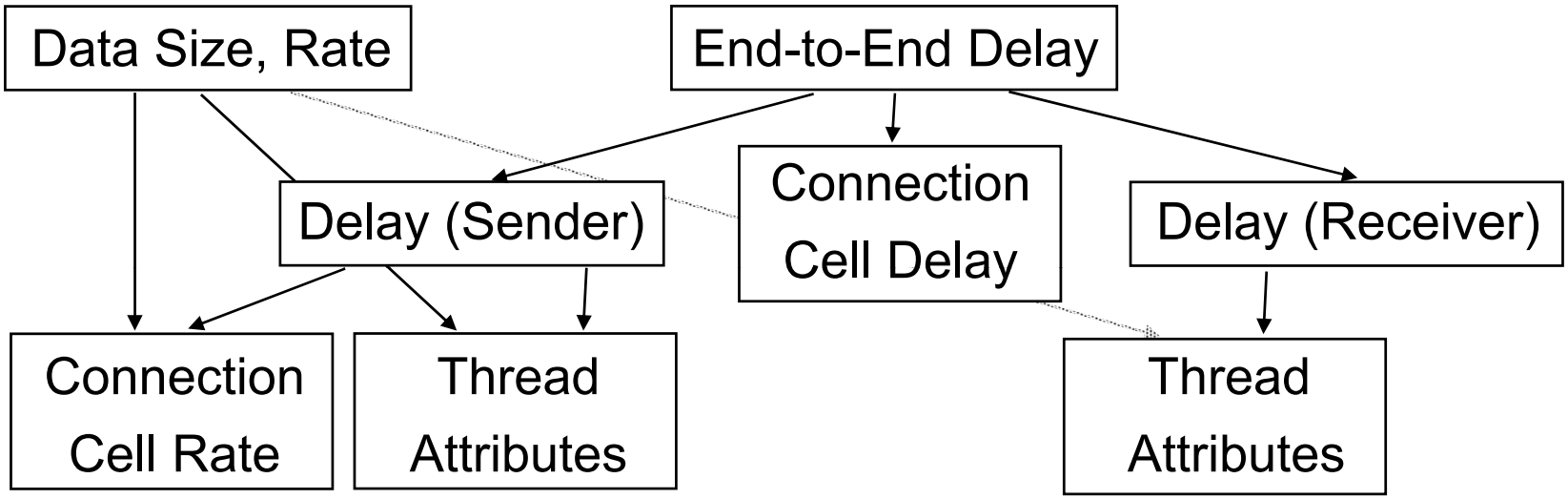
QoS Mapping – Operating System (1)

- Resource model and components:



QoS Mapping – Operating System (2)

- Steps:
  - Scale data sizes generated at the sender and partition the delay required for sender, network, and receiver (budget).
  - Determine connection parameters bandwidth and delay.
  - Derive protocol processing requirements as scheduling parameters (frequency and pacing).



QoS-Negotiation (1)

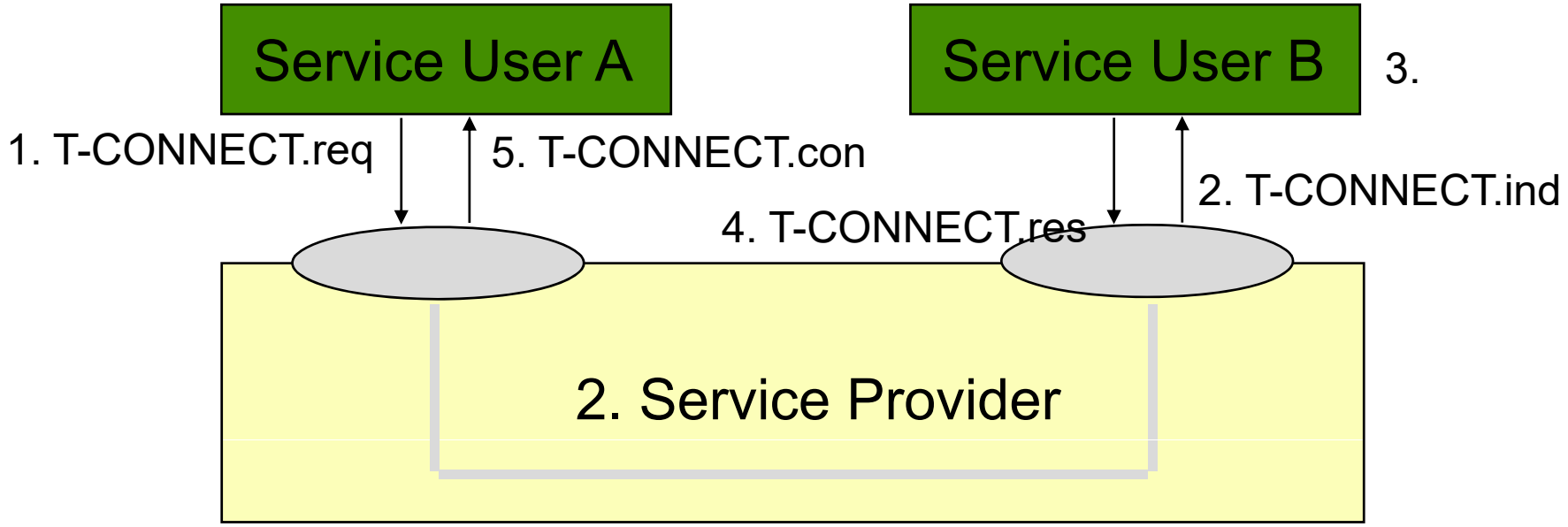
- Ideas:
  - Negotiating QoS values for an association (peer-to-peer),
  - Providing rules to optimize the exchange of information.
  - Prerequisites:
    - Definition of QoS parameters and
    - Specification of current QoS values requested.
- General methods:
  - Strengthen or weaken a proposed QoS parameter.
    - Increase or decrease a QoS parameter value.
  - Obey rules, who (entity) is allowed to change the value, *e.g.*, service provider, callee, or caller.
  - Activities: accept, reject, or change a proposal.
- Applied mechanisms:
  - Service primitives as defined above.
  - Additional handling procedures within protocols.
  - QoS management in end-systems.

QoS-Negotiation (2)

- Time of usage:
  - Static negotiation:
    - Happens during set-up of an association only.
    - QoS values and intervals are negotiated before the data transfer phase.
  - Dynamic negotiation:
    - QoS values and intervals may be negotiated during any phase of an association.
- Supported alternatives:
  - QoS value:
    - Setting of independent values or setting of dependent values
  - QoS interval:
    - Setting of independent intervals or setting of dependent intervals
  - Any mixture of values and intervals
- Negotiation choices:
  - Uni-lateral negotiation
  - Bi-lateral negotiation
  - Multi-lateral negotiation

Traditional QoS-Negotiation

1. QoS value proposed by the service user A.
2. Possible QoS modifications by the service provider.
3. Possible QoS modifications by the service user B.
4. QoS value finally selected by service user B.
5. QoS value indicated unchanged to service user A.



Comparison of QoS-Negotiation Schemes

Scheme	Applicable to cl	Applicable to co	Activities: Caller	Activities: Service Provider	Activities: Callee	Number of changes
Uni-lateral	yes	yes	Proposal	Acceptance Rejection	Acceptance Rejection	0
Bi-lateral	no	yes	Proposal	Acceptance Rejection	Acceptance Rejection Change	1
Tri-lateral	no	yes	Proposal	Acceptance Rejection Change	Acceptance Rejection Change	2
Multi-lateral	no	yes	Proposal	Acceptance Rejection Change	Acceptance Rejection Change	n

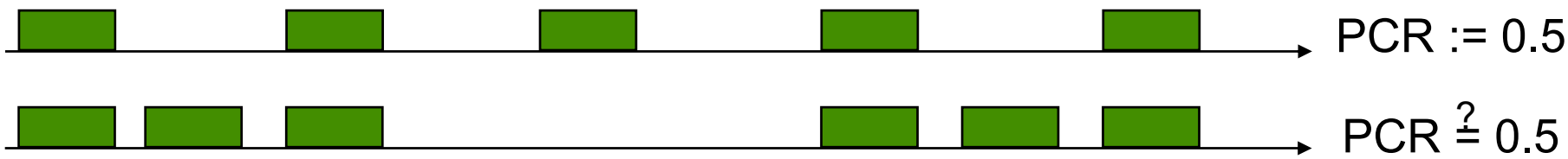
cl: connectionless co: connection-oriented

Admission Control

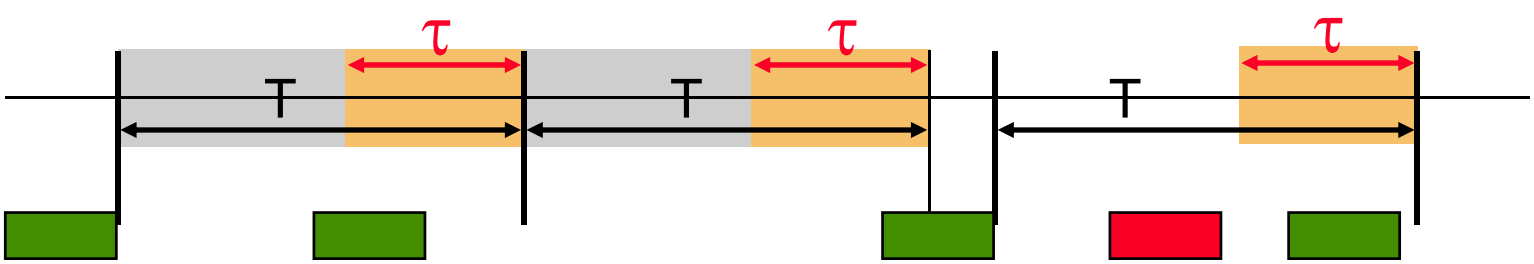
- Admission control assumptions:
  - Resources to be admitted have to be quantifiable, occupancy information need to be stored, and a test for admission must be feasible and available
    - Are sufficient resources available? NO: Reject admission request. YES: Proceed with resource reservation and association methods.
  - Examples:
    - Spatial requests: Memory, buffer, network capacity.
    - Timing requests: Schedulability, delay.
- Admission control performed during connection set-up time to:
  - Provide fairness between already existing and new flows,
  - Avoid overload situations of resources, memory or link bandwidth
  - Normally, admission control is based on average values, *e.g.*, for the cell rate, which implies the potential bottleneck for repetitive congestion due to bursty behavior of flows
- Advanced admission control serves well-utilized and optimally admitted resources

Admission Control – ATM Traffic Management

- Variable transfer delays due to “A” in ATM.
  - Clumping of cells.
  - Cell Delay Variation Tolerance (CDVT).

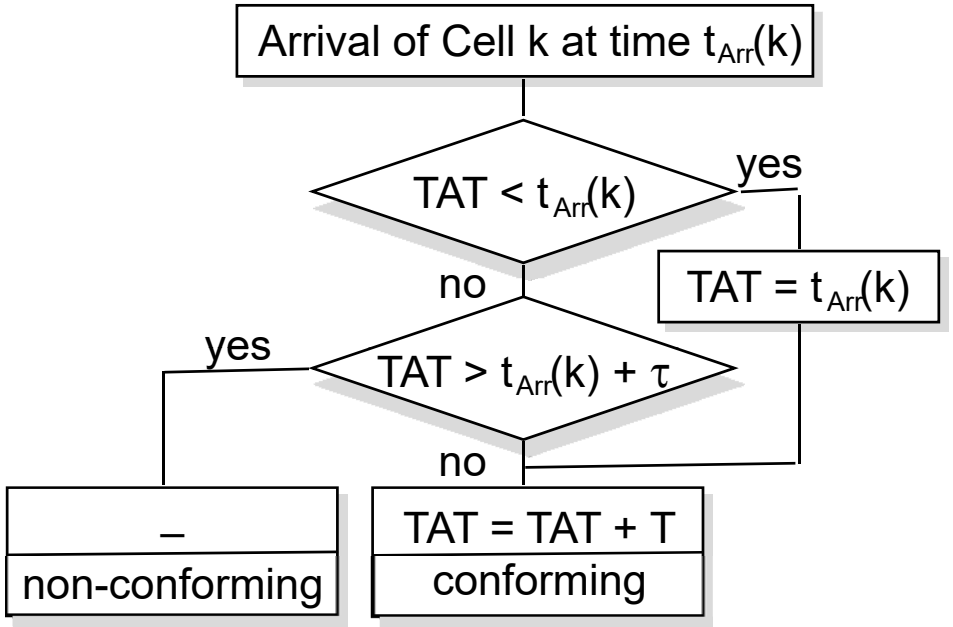


- Usage Parameter Control (UPC):
  - Check traffic characteristics by Generic Cell Rate Algorithm (GCRA) as a virtual scheduling based on leaky bucket.
  - Input parameter: T := incremental, τ := bound



ATM Admission Control (1)

- GCRA is defined independently by:
  - Virtual scheduling algorithm or Continuous state leaky bucket algorithm.
  - Parameter: T (incremental) for the rate of cells and τ (bound) for the degree of acceptable jitter: GCRA(T, τ).
  - GCRA calculates the Theoretical Arrival Time (TAT) which equals the nominal arrival time, if no jitter occurred.
  - Static algorithm due to parameters T and τ, and a traffic contract.



ATM Admission Control (2)

- GCRA(1/PCR, 0):
  Cell stream conforms to the Peak Cell Rate.
- GCRA(1/PCR, CDVT):
  Cell stream conforms to the PCR and Cell Delay Variation Tolerance due to variable delays between end-system and network access point.
- GCRA(1/SCR, BT):
  The Sustainable Cell Rate is an upper bound for the average cell rate over a long term period. The Burst Tolerance is used to gain resources and it is defined as: MBS = 1 + BT / (1/SCR – 1/PCR).
- GCRA(1/SCR, BT + CDVT):
  As above including the variable delays as above.

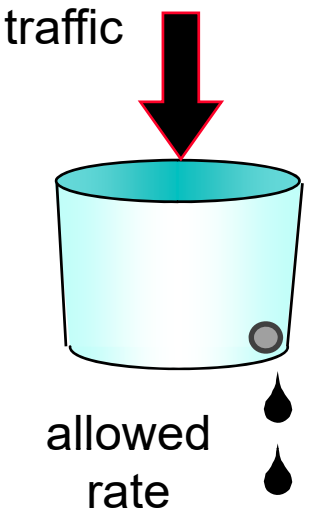
- Dynamic GCRA utilizes information from RM cells.

Need for Traffic Shaping

- Regulating traffic (Access Control) smoothes partially incoming/outgoing data to avoid congestion or supports billing.
- Example
  - Prerequisites: Gigabit network offering a bandwidth of 100 Mbit/s.
  - Flow A: Emits a single 100 Mbit packet every 1 second.
  - Flow B: Emits a single 1kbit packet every 10 microseconds.
  - ⇨ Consequences:
  - Transmission patterns are identical: 100 Mbit/s.
  - Buffer space in end-system. Problems with 100 Mbit packet in intermediate systems. Serialization within 100 milliseconds might interfere.
- Accepting a flow for a network access:
  - Provides a flow description of the planned behavior in terms of a flow shape.
  - Facilitates the decision whether a flow can be supported by a network.
- Policing a flow:
  - To determine whether the flow really behaves as initially stated.
  - Misbehaving can disrupt other traffic.

Traffic Shaping and Policing

- Idea of traffic shaping (Nutzungskontrolle):
  - Identification of a traffic shape (traffic pattern) for data flows.
    - Examples: isochronous, bursty, or asynchronous flows of data.
    - Definition of shape/pattern agreed upon (application-network contract)
- Use:
  - Specification of particular traffic characteristic (behavior) of a dedicated flow.
  - Including this information into traffic contract between application generating the flow and network, offering access control.
  - Allowing the network to supervise traffic characteristics of emitted data flows (policing).



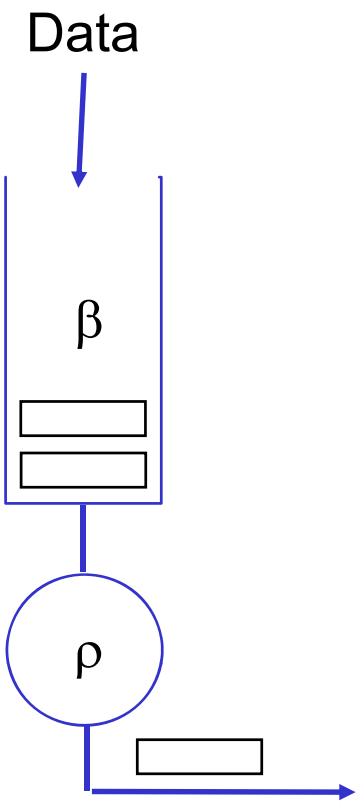
- Terminology:
  - Traffic shaping: The process of shaping traffic according to a scheme.
  - Traffic policing: Scheme to control and supervise incoming traffic into a network based on contracted traffic shapes.

Properties and Types of Traffic Shaping

- Properties
  - Range: Traffic shaping scheme able to describe a variety of traffic characteristics
  - Simplicity: Traffic shaping scheme provides easy approach to describe traffic char.
  - Qualification: Traffic shaping scheme shall be easy to police
  - Implementation: Traffic shaping scheme simple enough to be implemented in HW
- Types
  - Isochronous shaping:
    - Simplest scheme, smoothing all traffic to constant bit rate traffic
      - Examples: Leaky Bucket, (r,T)-Smooth Traffic
  - Isochronous shaping with priorities:
    - Keeping the scheme simple, limited support of variable bit rate traffic
      - Example: Leaky Bucket with Priorities
  - Bursty shaping:
    - Supporting a richer range of traffic patterns, complex approach
      - Examples: Token Bucket, Token Bucket with Rate Control

Leaky Bucket (1)

- Each data flow owns a leaky bucket with size β.
  - Maximum size limits the amount of data.
  - Additional data will be discarded.
- To be sent data are placed in the bucket.
  - Data is meant to be in “cell size”.
- Cells leave the bucket as they are sent via the network with rate ρ.
  - The rate is enforced at the bucket bottom.



- Traffic pattern description:
  - Rate ρ.

Leaky Bucket (2)

- Consequences for the traffic pattern:
  - Changing bursty traffic patterns into isochronous ones.
  - Smoothing: 1 cell per $1/\rho$ time unit.
  - Data injection always slower that rate ρ.
  - Size β bounds delay of to be emitted data: waiting time in the maximum-sized bucket is limited.

- Implementation:
  - Simple FIFO buffer sufficient.
  - Single timer clocking at every $1/\rho$ time unit.

- Policing:
  - If two cells are ever send together in less time than $1/\rho$, the flow is violating the traffic pattern.

Leaky Bucket with Priorities (1)

- Support of variable bit rate flows by priorities.

- Solution:
  - Bucket size exceeding data are assigned a lower priority.
  - Under heavy load these data will be discarded:
    - Either in the end-system or
    - In an intermediate system.
  - Marking with priorities is done by:
    - Application or
    - Network.

- Further examples for applied priorities:
  - ATM supports priorities by the CLP bit.
  - MPEG defines implicit priorities by B, I, or P frames.

Leaky Bucket with Priorities (2)

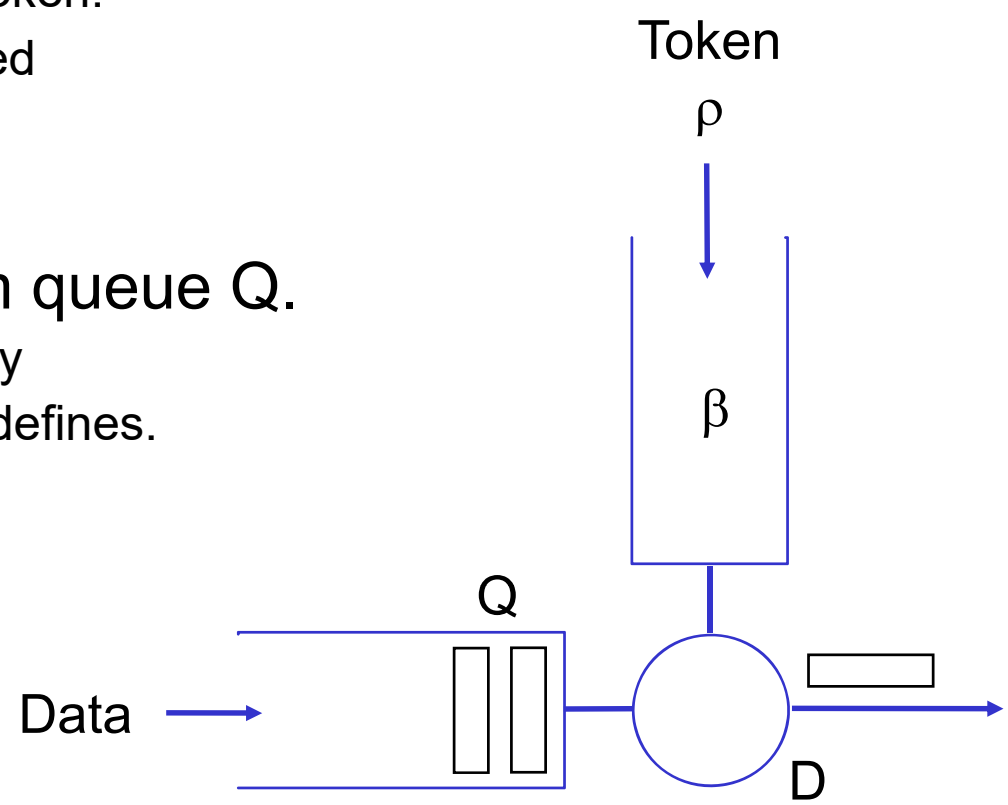
- Concerns:
  - Guarantee of traffic is difficult. Investigations are still missing to identify how much bandwidth, *e.g.*, 50% (?), is needed for low-priority traffic to be useful.
  - cf. implementation below.

- Implementation:
  - Problems are encountered. Data is stored in FIFOs, accessible at the front or end only. When does the controller discard low-priority data? Anticipating overloads or blocking the transmission queue?

- Policing:
  - User friendly. In case of under-load, cells will be transmitted, in cases of overload they will be discarded.

Token Bucket (1)

- Each data flow owns a token bucket with size β.
  - Bucket manages device D to regulate the flow.
  - Maximum size limits the amount of token.
  - Arriving tokens with rate ρ that exceed the bucket size β will be discarded.
- To be sent data are taken from queue Q.
  - To transmit data, D removes as many tokens from β, as the length of data defines.
- Traffic pattern description:
  - Long term/average rate ρ.
  - Peak rate $\beta + \rho * \tau$ during interval of τ time units.



Token Bucket (2)

- Effects are different compared to the leaky bucket. Assume sending data of size *b* tokens ($b < \beta$) :
  - Token bucket is full, $B = \beta$ (B is the number of tokens in β):
    Data are sent and B tokens are removed.
  - Token bucket is empty, $B = 0$:
    Data has to wait until *b* tokens arrived in the bucket.
  - Token bucket is partially full. Two cases are considered:
    - $B \geq b$: Data is sent immediately.
    - $b > B$: Data must wait until $B - b$ remaining tokens arrived in the bucket.

- Token bucket permits a bounded burstiness.
  - During an interval of τ time units, no more data than
    $\beta + \tau * \rho$ tokens are sent.

Token Bucket (3)

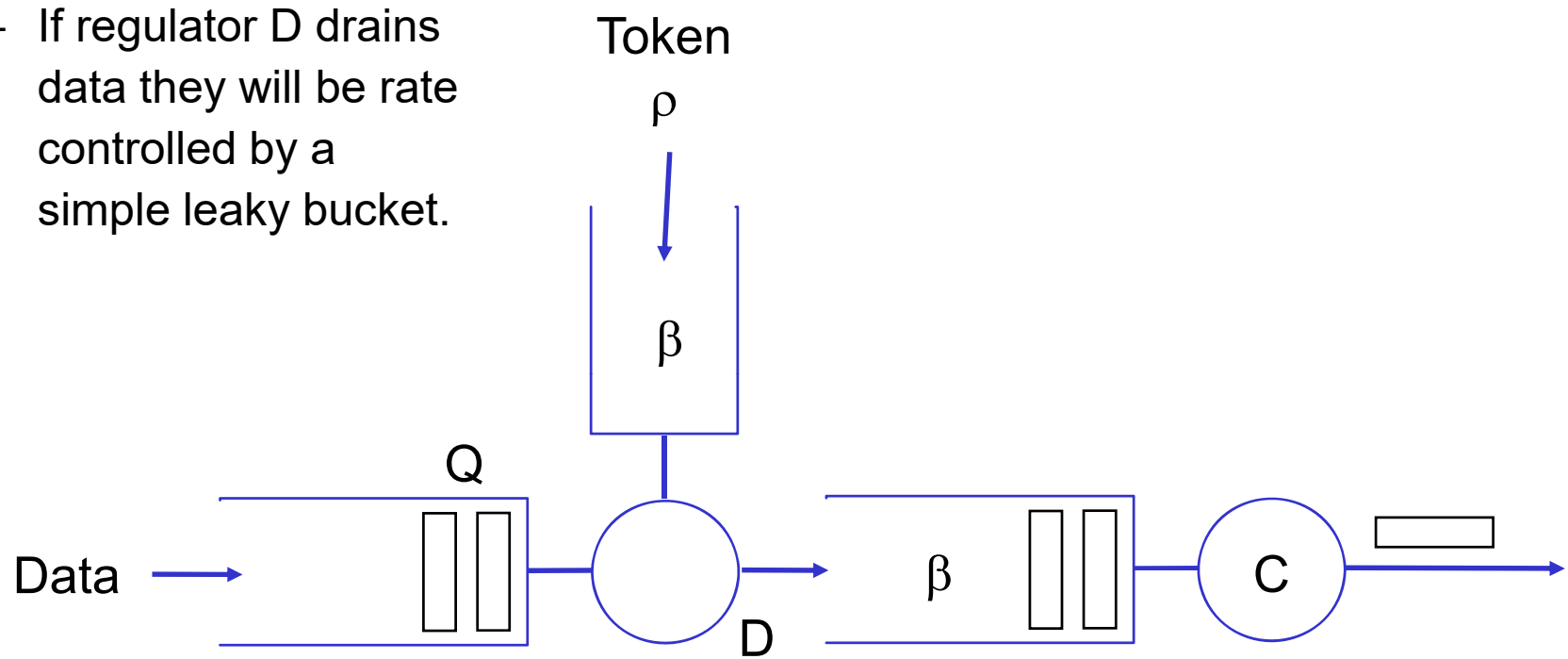
- A token bucket offers a constant long-term transmission rate, defined by the token arrival rate ρ.

- Discarding and priorities:
  - Exceeding tokens are discarded and the problem of managing the transmission queue Q is left to the flow.
- Implementation:
  - Simple FIFO buffer sufficient.
  - Single timer clocking at every $1/\rho$ time unit to add tokens to the bucket and a counter for counting tokens to be used.
- Policing:
  - Difficult: In any period of time the flow may exceed ρ by β tokens. Checking for the interval τ only, allows to cheat.

Token Bucket with Rate Control (1)

- Token bucket introduces bursty data up to the size β of the token bucket.

- For minimizing harms of long bursts two choices exist:
  - Limiting size of β, being too restrictive.
  - If regulator D drains data they will be rate controlled by a simple leaky bucket.



Token Bucket with Rate Control (2)

- Effects:
  - The final transmission rate C has to be significantly greater (faster) than token arrival rate ρ: C >> ρ.
  - This allows for burstiness as within the token bucket, but interleaves long bursts by pacing data according to rate C.
- Implementation:
  - Complex.
  - Each flow requires one timer and one counter for every bucket, in total two timers and two counters.
- Policing:
  - Remains hard, except for confirming that the data rate does not exceed C.

Comparison of Traffic Shaping Schemes

Scheme	Transmission Rate ave.	Transmission Rate peak	Data Size	Data Disc-arding	Traffic Shape	Burst Size	Delay	Prio-rity	Idle Case
Leaky Bucket	ρ	ρ	$< \beta$	$> \beta$	isochronous per $1/\rho$ time	β	$< \beta$ bounded	no	no data
(r,T)	Tc/ρ per T bit times	Tc/ρ	$< r$	–	plesiochronous	r	–	no	no data
Leaky Bucket with Priorities	ρ	ρ	$< \beta$	$> \beta$	isochronous per $1/\rho$ time	β	$< \beta$ bounded	yes	no data
Token Bucket	ρ within interval τ	$\beta + \tau * \rho$	oo	–	bounded burstiness	$\beta + \tau * \rho$	–	no	no data
Token Bucket with Rate Control	ρ	C	oo	–	bounded burstiness and plesiochronous	$\beta + \tau * \rho$ smoothed by C	–	no	no data

Summarized Advantages and Drawbacks

Scheme	Advantages	Drawbacks
Leaky Bucket	simple to implement and police	isochronous traffic patterns only, constant bit rate data flow only
(r,T)	simple to implement and police	no cell longer than *r* bits
Leaky Bucket with Priorities	“simple” to implement and police	selectively discarding packets, initial bandwidth reservation for low-priority traffic necessary
Token Bucket	accurate description of traffic pattern, includes high burstiness	policing, FIFO access, anticipating heavy load situations, high burstiness, overflowing data queue
Token Bucket with Rate Control	accurate description of traffic pattern, includes controlled burstiness, allows for resource allocations	hard policing, complex implementation overflowing data queue

QoS Monitoring

- QoS monitoring extracts information on current service behavior
  - Event-driven monitoring feasible
  - Continuous monitoring with bigger effort
  - Prerequisite: Monitored QoS information stored
    - Internal data structures (for QoS methods) or Global structures, *e.g.,* MIB

optimally without any overhead

- Five steps required for performing QoS monitoring
  - QoS identification: Define to be measured QoS parameters at the corresponding service access points and relevant parameters accessible within the communication subsystem
  - QoS tools: Determine the SW or HW mechanisms to be used for collecting and extracting QoS information
  - QoS measurement: Placement of QoS tools within the end-system, the protocol, and further relevant components
  - QoS calculation: Compute current values for identified QoS parameters based on measured parameters
  - QoS supervision: Initiate actions based on negotiated and contracted QoS parameter limits, thresholds, or bounds

QoS Enforcement

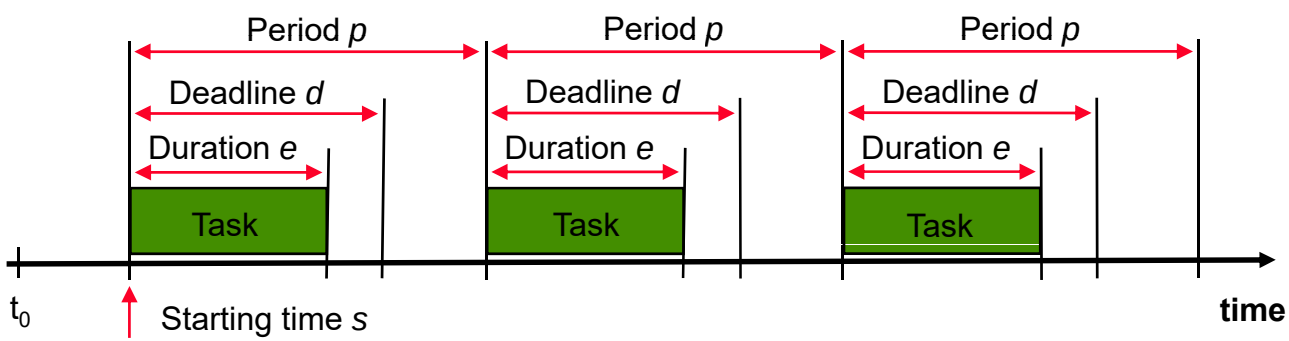
- QoS enforcement, based on monitored QoS, requires mechanisms to enforce the QoS contract and to maintain current QoS as close as possible, while controlling and scheduling end-system resources
  - *E.g.*, process manager to assign processes to the CPU

- Fulfilling multimedia application requirements demands real-time support for continuous data
  - Traditional real-time: high security and fault tolerance
  - Multimedia systems are real-time systems with less security and fault tolerance needs, but a periodic behavior of the traffic or the task

⇨ Necessity of scheduling algorithms

Scheduling and its Modeling

- Problem: Sharing of utilized resources in time and space (*e.g.*, link, processor, memory)
- A scheduler defines disciplines for resource sharing
  - Timeliness
  - Algorithms
  - Load balancing or traffic control
- Tasks are basic units of schedulability
  - Preemptive: Tasks may be interrupted from a higher priority task at any time
  - Non-preemptive: Tasks are never interrupted. A task only deliberately releases control of the CPU. Tasks of higher priority need to wait (priority inversion)
- Periodic tasks are modeled by ($0 < e \leq d \leq p$):

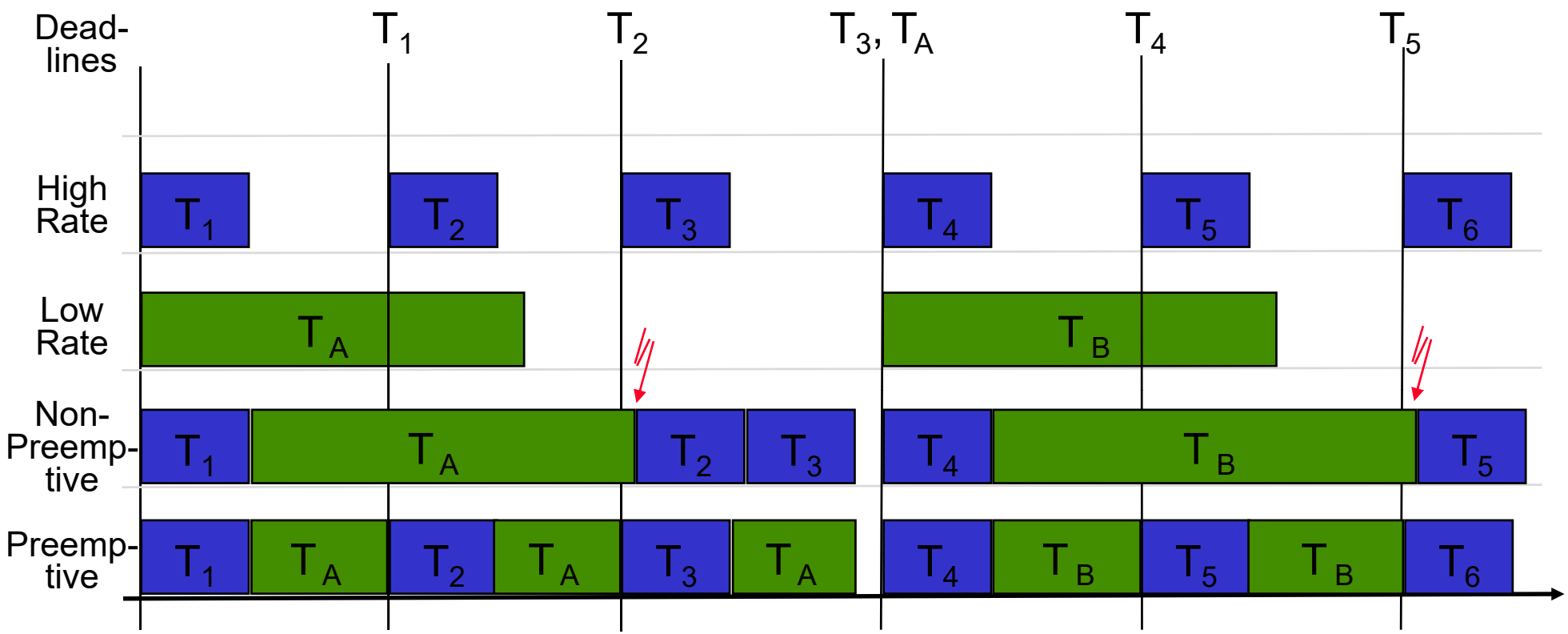


Scheduling Goals and Example Algorithms

- Ease of implementation
  - Little time per scheduling decision at high link speeds
  - Independent of number of active connections: O(1)
  - Implement the discipline in hardware, problem: state
- Fairness and protection
  - Fair share of link capacity, output buffers to each connection determines a desirable property for best-effort traffic
  - Guaranteed connections pay a usage-based price per resource
  - Misbehaving connections should not affect others
- Performance bounds
- Ease and efficiency of admission control
- Example scheduling algorithms include
  - First-Come-First-Serve (FCFS) or First-In-First-Out (FIFO),
  - Earliest Deadline First (EDF),
  - Weighted Fair Queuing (WFQ), or
  - Virtual Clocks

Scheduling – Preemption

- Difference of preemptive and non-preemptive task scheduling is presented in this example
  - Violations of the deadline are marked by ⚡.



Earliest Deadline First Algorithm (1)

- EDF: Earliest Deadline First scheduling algorithm
- EDF calculates at the arrival time of a new task the new schedule, while preempting the current task and scheduling the task with the next approaching deadline first, while assigning every task a certain priority
  - EDF is dynamic due to the new calculation of a schedule each time a new tasks arrives
  - EDF is optimal, since it calculates a schedule if it exists
  - EDF optimally utilizes the processor by 100%
  - EDF overhead is due to recalculating every task priority in the worst case

Earliest Deadline First Algorithm (2)

- An EDF extension schedules tasks, but aborts tasks that will not meet the required deadline

- A further extension: tasks may be subdivided into mandatory and optional parts. In this case the scheduling takes place by assuming the mandatory part as important
  - Coding of pictures by 16 bit resolution per pixel needs in overload situations the first 8 bit only (mandatory part)
  - Reduction of processing time required for coding
  - Reduction of picture quality due to less coding
  - QoS negotiation may be required in this case

Rate Monotonic Algorithm (1)

- RM: Rate Monotonic scheduling algorithm

- RM assigns a fixed priority to every task depending on their period (small period length ⇨ high priority).
  This happens once at connection set-up time for all tasks being served

  – RM is optimal, since RM calculates a schedule
  – RM is static, since the schedule is being calculated once without any rearrangements
  – RM is priority-driven, depicting a relevance of tasks relative to other tasks
  – RM serves periodic and preemptive tasks

Rate Monotonic Algorithm (2)

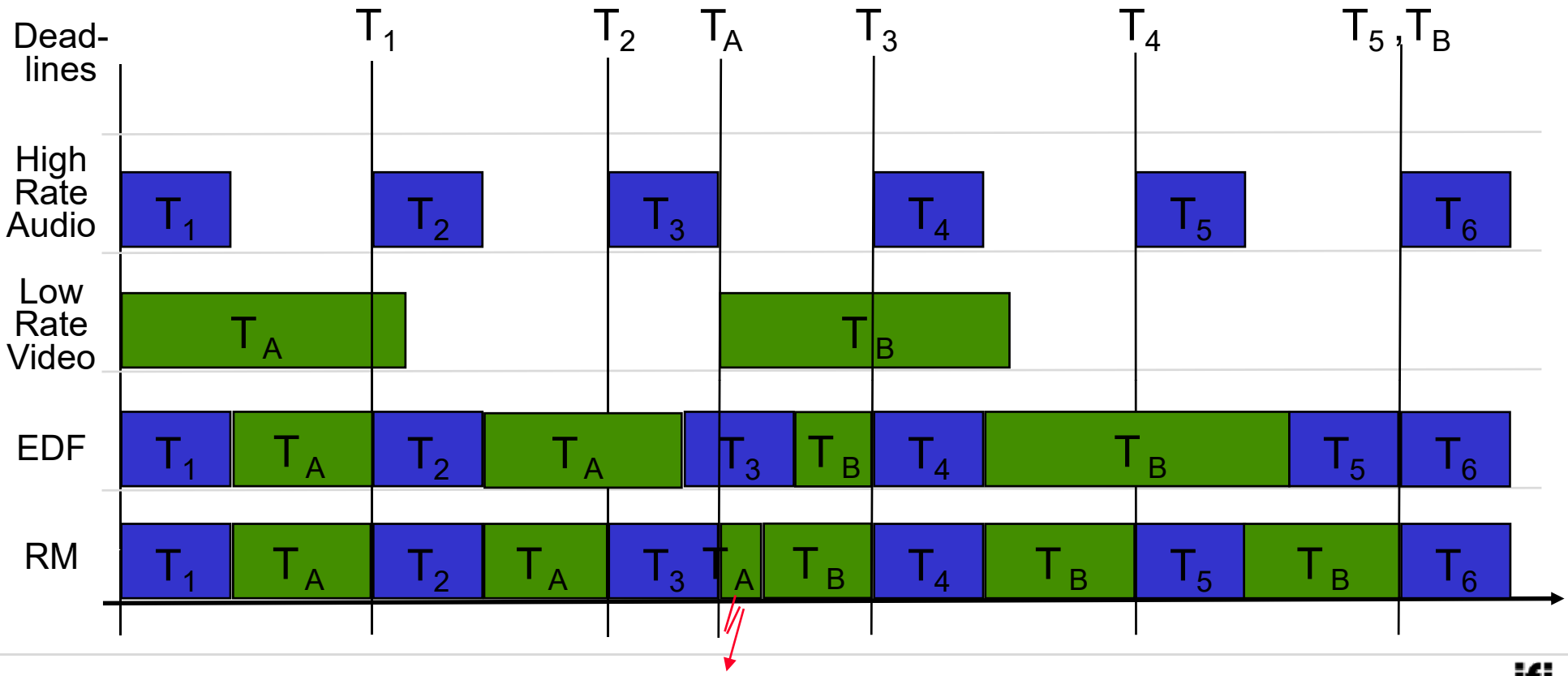
- RM assumes five prerequisites
  - Task arrivals are periodic
  - Tasks in a single flow of data are of pure sequential nature
  - All tasks are independent from each other
  - A maximum processing time for every task is available
  - Any non-periodic task does not require a deadline
- Utilization for RM
  - Assuming that period *p* equals deadline *d* (*p* = *d)* the processor utilization equals

$$U(m) = m * (2^{1/m} - 1) > \sum_{j=0}^{m} \frac{e_j}{p_j}$$

  - A lower bound is determined by $\ln(2) = 0.69$
- Deadline guarantees are based on this utilization

Comparison of EDF and RM

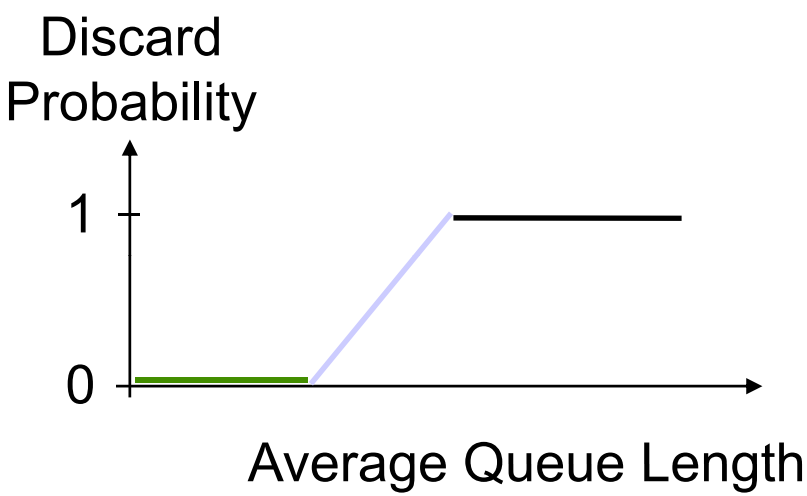
- Deadline violations (⚡) in RM may not have occurred within the EDF algorithm
  - Number of context switches in RM is much higher



Random Early Discard (RED)

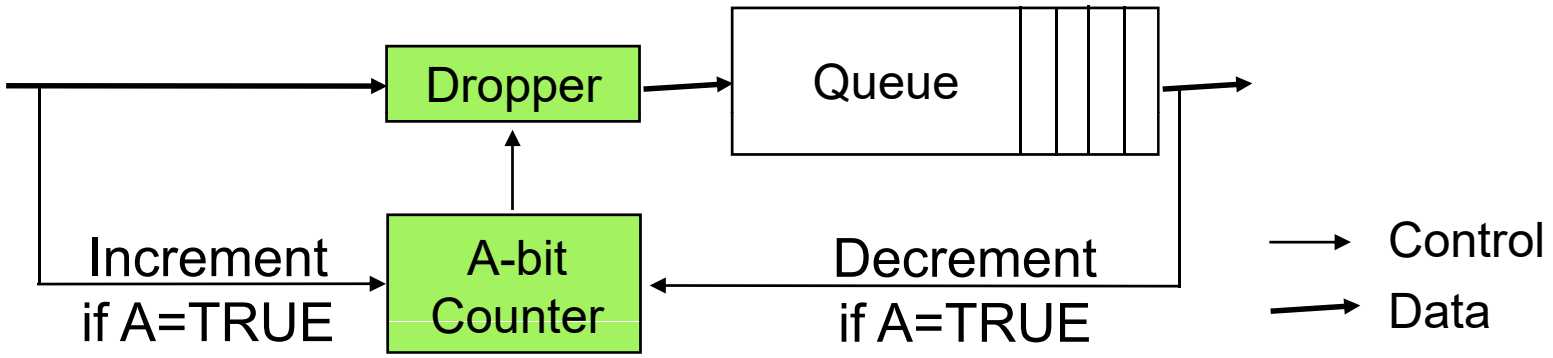
- Passive queue management
  - Dropping packets at the tail of a queue.
  - Problems
    - Prioritization of queue by certain flows,
    - Long delays for filled-up queues, and
    - Impossibility to absorb bursts

- Active queue management
  - Random Early Discard (RED).
  - Features:
    - Estimation of queue length
    - Discard probability depends on average queue length



RED with In and Out (RIO)

- Active queue management
  - Application of RED with two different disciplines for in-service (A-bit) and out-of-service packets
    - Out-dropper discards aggressively
    - In-dropper is based on in-service packets per queue
  - Scheme avoids dropping of in-service packets



Pre-allocated Scheduling: Max-min Fairness

- Max-min fairness maximizes the minimum share of sources, whose demand is not fully satisfied:
  - Users with small demands achieve smaller shares.
    - Resources are allocated in order of increasing demand.
    - No source gets a share larger than its demand.
    - Sources with unsatisfied demands get an equal share.
  - n sources S demand x_i each and resource's capacity is C.
    - Ordering: $x_1 \leq x_2 \leq ... \leq x_{n-1} \leq x_n$.
    - S_1 := C/n and C/n - x_1 may be available (for S_1 unused).
    - Even distribution to n-1 sources: C/n + (C/n - x_1)/(n - 1)
  - Enhancement by weights w_i per source. Allocation per increasing demand (normalized by weight) and unsatisfied sources receive shares proportional to their weights.

Max-min Fairness – Examples

- n=4 sources with demands x_i 2, 2.66, 4, and 5, and resource capacity C=10. Shares are denoted by c_i.
  - c_1 = C/n = 10/4 = 2.5: 2.5 > 2.
    c_1 = 2.0. Rest of 0.5 unused from S_1.
  - Even distribution to S_2, S_3, and S_4: 0.5/(n -1) = 0.5/3 = 0.16:
    c_2 = 2.5 + 0.16 = 2.66. No unused rest remains.
  - c_3 = 2.5 + 0.16 = 2.66. No unused rest remains.
  - c_4 = 2.5 + 0.16 = 2.66. No unused rest remains.
  - C = $c_1 + c_2 + c_3 + c_4$ = 2.0 + 2.66 + 2.66 + 2.66 = 10.

- Weighted approach with n = 4 sources with demands x_i 4, 2, 10, 4, weights w_i 2.5, 4, 0.5, 1, C=16.
  - Normalization: w_i 5, 8, 1, and 2. Now n=16. As above.
  - Fair shares c_i 4, 2, 6, 4.

Scheduling – Multimedia Aspects

- Multimedia scheduling considers two conflicting goals
  - Non real-time processes shall not suffer due to critical processes
  - Time-critical processes shall never experience priority inversion

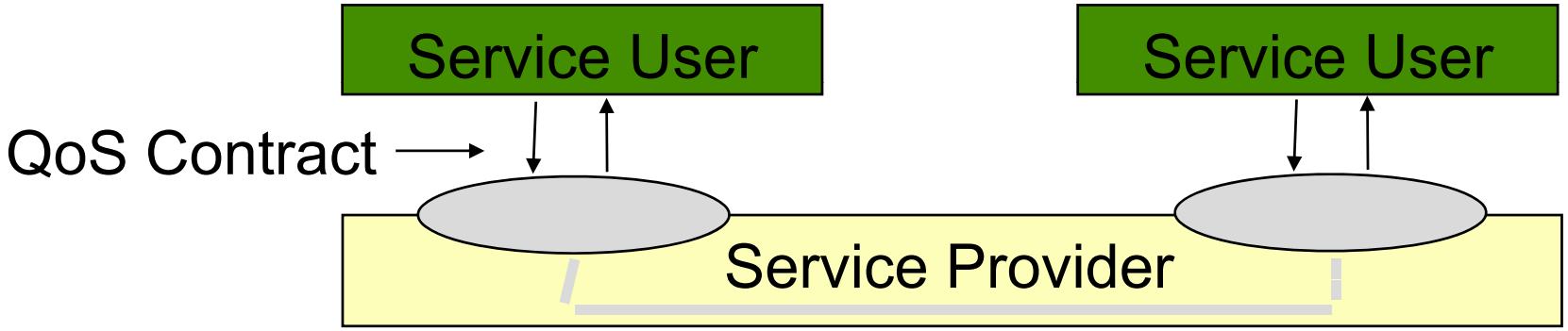
- Two basic approaches for scheduling algorithms are used often within different variations for multimedia operating systems
  - Earliest Deadline First (EDF) and
  - Rate Monotonic (RM).

- Initially multimedia operating system examples:
  - *E.g.*, Nemesis, SUMO, YARTOS, WANDA, Real-time Mach

- Today almost all operating systems provide scheduling for data streams and resource over-provisioning

QoS Contracts

- QoS contracts are part of the QoS management, but are a direct output of QoS methods.
  - They comprise the negotiated and contracted QoS parameter values for a particular flow of data.
  - They propose the agreement on QoS for the service user(s) and the service provider.
  - They are the result of QoS methods applied during the association establishment, *e.g.*, QoS mapping, QoS negotiation, and resource reservation and association.



References

- A. Campbell, G. Coulson, D. Hutchinson: *A Quality-of-Service Architecture;* ACM Computer Communication Review, Vol. 24, No. 2, April 1994, pp 6 – 27.
- Open Systems Interconnection, Data Management and Open Distributed Processing: *Information Technology – Quality of Service – Framework;* ISO/IEC JTC1/SC21 N9680, (Final) CD 13236.2, July 1995.
- M. Ritter, P. Tran-Gia: *Mechanismen zur Steuerung und Verwaltung von ATM-Netzen, Teil 1: Grundlegende Prinzipien;* Informatik-Spektrum, Vol. 20, No. 5, April 1997, pp 216 – 224.
- R. Stankiewicz, P. Cholda, A. Jajszczyk: QoX: What is it really? IEEE Communications Magazine, Vol. 49, No. 4, pp. 148-158, April 2011.
- R. Steinmetz: *Analyzing the Multimedia Operating System;* IEEE Multimedia, Vol. 2, No. 2, Spring 1995, pp 68 – 84.
- B. Stiller: *Quality-of-Service Methods for Managing Communicating Applications;* Journal on Integrated Computer-aided Engineering, IOS Press, Vol. 6, No. 2, February 1999, pp 159 – 169.
- X. Xiao, L. M. Ni: *Internet QoS: A Big Picture;* IEEE Network Magazine, Vol. 13, No. 2, March/April 1999, pp 8 – 17.