Quality of Service (QoS) over IP Networks

Luca Veltri

(mail.to: luca.veltri@unipr.it)

Corso di Reti di Telecomunicazioni C, a.a. 2008/2009

http://www.tlc.unipr.it/veltri

Introduction to QoS (1/3)

- Definition: A network supports Quality of Service (QoS) when it provides:
  - **Performance Assurance:**
    Gives guarantees on some performance parameters (e.g. throughput, delay, jitter)
  - **Service Differentiation:**
    Provides different services to different applications according to their requirements

Introduction to QoS (2/3)

- IP does not natively support QoS:
  - **IP offers a simple point-to-point, unreliable, connectionless delivery service (Best Effort service)**
  - **no constraints are imposed on user generated traffic (i.e. no admission control and no traffic policing are performed)**
- Current practice:
  - **common over-subscription, that implies too many users with respect to available resources**
  - **Internet characterized by a strong heterogeneity of inter-connected networks (e.g. throughput can vary within several magnitude orders)**

Introduction to QoS (3/3)

- In order to provide QoS over IP Networks the IETF has developed two major frameworks:
  - **The Integrated Services Architecture (IntServ)**
  - **The Differentiated Services Architecture (DiffServ)**
- IntServ and DiffServ relies on proper mechanisms, both in the User Plane and in the Control Plane:
  - **Data Path Mechanisms**
  - **Control Path Mechanisms**

QoS Basics: Flows

- Flow:
  - **A distinguishable stream of related datagrams that results from a single application**
  - **It is commonly defined by the 5-tuple:**
    - Source and Destination IP Address
    - Protocol Type (e.g. UDP/TCP)
    - Source and Destination Ports
- Service Differentiation and Performance Assurance can be provided per-flow (finer granularity) or per aggregate (coarser granularity)

QoS Basics: Token Bucket

- Token Bucket:
  - **It is a model that describes the characteristics of a flow or an aggregate**
  - **It simply gives an upper bound to the emitted traffic**
  - **A flow (aggregate) conform to a Token Bucket if for any given time interval T the number of transmitted bytes is less than or equal to:**

    $$A(T) = b + rT$$

    where:
    - b = Burst size (bytes)
    - r = Average long term rate (bytes/sec)
  - **or more precisely:**

    $$A(T) = \min [M+p \cdot T, b+r \cdot T]$$

    where:
    - p = peak rate (bytes/s)
    - M = max. datagram length
    - m = min. datagram length

Data Path and Control Path Mechanisms

Data Path Mechanisms (1/7)

- Basic building blocks on which IP QoS is built
- Actions to be taken on a per packet basis in order to enforce different levels of service:
  - **Basic Packet Forwarding Operations**
    - Classification
    - Metering
    - Marking
    - Shaping
    - Policing
  - **Queue Management**
  - **Scheduling Mechanisms**

Data Path Mechanisms (2/7)

- Classification:
  - **selects packets and determines which flow or class they belong according to specified rules**
  - **classification is based on the content of some portion of the packet header:**
    - Multi Field (MF) Classification:
      - Matching is based on a tuple in the packet header
      - Processing intensive operation
      - Fine granularity
    - Behavior Aggregate (BA) Classification
      - Matching based on a single field in the packet header
      - Simpler and faster than MF Classification
      - Coarser granularity

Data Path Mechanisms (3/7)

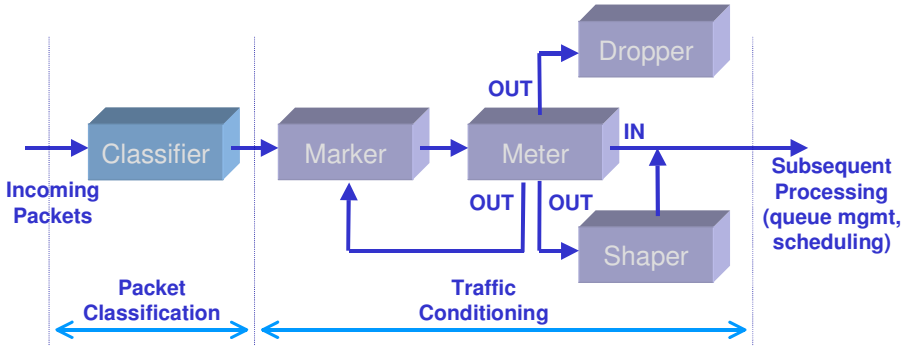
- Marking:
  - **Marks a packet for differentiated treatment later**
  - **Packets to be marked are selected by a classifier**
- Metering:
  - **Measures the temporal properties of a traffic stream against a traffic profile (e.g. a Token Bucket)**
  - **In profile packets are put in different service queues for further processing**
  - **Out of profile traffic can be remarked, dropped or delayed in a shaper**

Data Path Mechanisms (4/7)

- Shaping:
  - **Delays packets of a traffic stream in order to bring it into compliance with a specified traffic profile**
  - **An ideal shaper can be thought as an infinite length buffer followed by a Token Bucket**
- Policing:
  - **Drops out of profile packets**
  - **It can be implemented by a shaper with a zero length buffer**

Data Path Mechanisms (5/7)

- Basic Packet Forwarding Blocks:



Data Path Mechanisms (6/7)

- Queue Management Mechanisms:
  - **Control and avoid Network Congestion**
  - **They affect Packet Loss and Jitter**
  - **Traditionally packets are dropped when the queue is full:**
    - Tail Drop ⇒ Arriving packet is dropped
    - Drop Front ⇒ Packet in the queue from the longest time is dropped
    - Random Drop ⇒ Randomly chosen packet is dropped
  - **Problems:**
    - Unfairness (a flow can monopolize queue space)
    - Interaction with TCP congestion control (variable throughput and delay)

Data Path Mechanisms (7/7)

- Scheduling mechanisms:
  - **Choose the next packet to be transmitted**
  - **A proper scheduling algorithm can guarantee:**
    - Delay Control
    - Link Sharing (Rate control)
  - **Can be performed on a per-flow or aggregate basis (or a combination of them ⇒ Hierarchical scheduling)**
  - **There are several Scheduling Disciplines:**
    - FIFO (First In First Out)
    - Priority Scheduling
    - WFQ (Weighted Fair Queuing)
    - EDF (Earliest Deadline First)

Control Path Mechanisms (1/3)

- Concerned with configuration of network nodes in order to:
  - **distinguish packets entitled to receive special treatment**
  - **manage resources according to specified rules**
- They are NOT applied on a per packet basis:
- The main ones are:
  - **Admission Control**
  - **Policy Control**

Control Path Mechanisms (2/3)

- Admission Control:
  - **Used to determine whether a new traffic stream can be admitted without impacting previously granted QoS requests**
  - **Requires that traffic and QoS characteristics as well as current state of the network are known**
  - **Three basic approaches:**
    - Deterministic:
      A-priori estimation based on the worst case (very conservative)
    - Statistic:
      A-priori estimation based on the average case (better utilization)
    - Measurement based:
      Based on the current measurement of some parameters

Control Path Mechanisms (3/3)

- Policy Control:
  - **Regulates access to network resources based on administrative criteria**
  - **Policies determines which users/hosts/applications should have access to which resources/services and under what conditions**
  - **Policy rules are usually in the form:**
    "If a certain condition occurs then take certain actions"
  - **Usually Policies are applied by means of a proper Policy Architecture**

The Differentiated Services Architecture

Differentiated Services

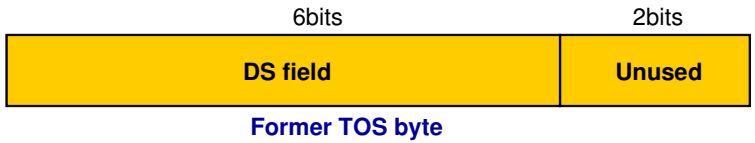
- QoS capable architecture born as an alternative to IntServ in order to achieve scalability (RFC 2475)
- The main DiffServ characteristics are:
  - **Aggregate based**
  - **Static resource allocation based on Service Level Agreements (SLAs)**
  - **Absence of signaling**
  - **Absence of Admission Control**
  - **Stateless approach**

Differentiated Services

- The DiffServ architecture aims to provide scalable service differentiation in the Internet
  - **A service is defined as *"the overall treatment of a defined subset of customer's traffic within a domain, or end-to-end"***
- A service is composed of 4 elements:
  - **Per Hop Behaviors (PHBs) within the core of the network**
  - **Traffic Conditioning rules enforced at the boundaries**
  - **Provisioning strategies**
  - **Billing models**
- PHBs are the basic building blocks for service provisioning
  - **A small set of PHBs can be used to provide quite different types of services if properly coupled with Traffic Conditioning rules and resource provisioning within the network**

DiffServ Terminology

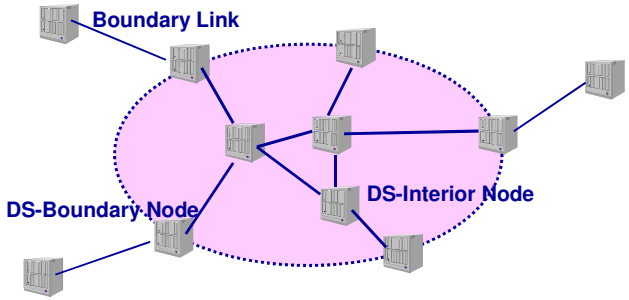
- DS field
  - **the 6 MSB of the former IPv4 TOS octet (or the IPv6 Traffic Class octet). The bits of the DS field encode the DS codepoint, while the remaining bits are currently unused**



- DS codepoint (DSCP)
  - **a specific value contained in the DS field and used to select a PHB**

DiffServ Terminology (cont.)

- DS-domain
  - **a contiguous set of nodes which operate with a common set of service provisioning policies and PHB definitions**



DiffServ Terminology (cont.)

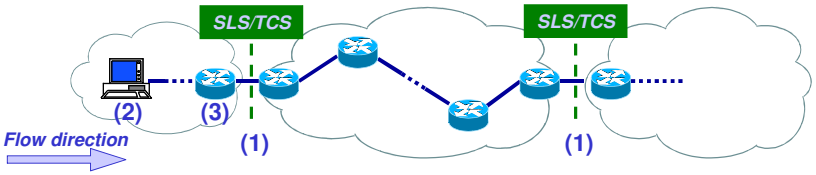
- Service Level Specification (SLS)
  - **a service contract between a customer and a service provider that specifies the forwarding service the customer should receive. A customer may be a user organization (source domain) or another DS domain (upstream domain)**
  - **a SLS may include traffic conditioning rules which constitute a TCS in whole or in part**
- Traffic Conditioning Specification (TCS)
  - **an agreement specifying classifier rules and any corresponding traffic profiles and metering, marking, discarding and/or shaping rules which are to apply to the traffic streams selected by the classifier**

Static vs Dynamic SLS

- Static vs Dynamic Service Level Specification:
  - **Static SLS:**
    - Static SLSs are the norm at present time
    - They are agreed after negotiation between human agents
    - They can be periodically renegotiated (days… weeks …)
    - The SLS can specify a time varying services (times of day)
  - **Dynamic SLS:**
    - Dynamic SLSs may change frequently, as a result from variations in offered traffic load or from changes in pricing offered by the provider
    - Dynamic SLSs change without human intervention and require an automated agent and protocol (e.g. “Bandwidth Broker”)

Differentiated Services Model

- End-to-end QoS provisioning in DiffServ:



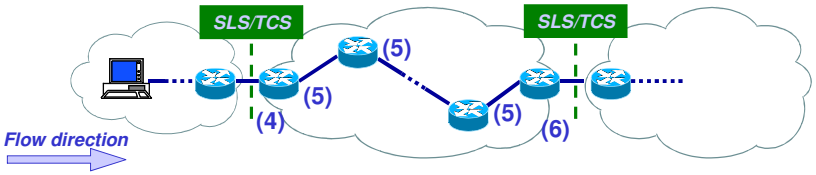
(1) Negotiation of SLS/TCS between peering domains

(2) Premarking in the source domain (e.g. per app/host basis)

(3) Egress boundary DS node of source domain applies traffic conditioning to ensure SLS/TCS compliance, hence causing possible re-marking, dropping and shaping

Differentiated Services Model

- End-to-end QoS provisioning in DiffServ (cont.):



(4) Classification and Conditioning according to SLS/TCS Assignment to a BA (DSCP setting)

(5) Forwarding according to the PHB associated with the assigned DSCP

(6) Iteration of (3)-(5) for each downstream domain

Expedited Forwarding PHB

- Expedited Forwarding (RFC 2598)
  - **The EF PHB can be used to build a low-loss, low-latency, low-jitter, assured bandwidth end-to-end service through DS domains**
  - **Queuing must be avoided: aggregate maximum arrival rate is less than that aggregate's minimum departure rate**
    - Nodes must be configured so that the aggregate has a "well-defined" (i.e. independent of other traffic on the node) minimum departure rate
    - The aggregate must be configured (via policing and shaping) so that its arrival at any node is always less than the configured minimum departure rate

Expedited Forwarding PHB

- Expedited Forwarding (cont.)
  - **The departure rate of aggregate must equal or exceed a configurable rate**
    - The EF traffic should receive this rate independent of other traffic
    - The rate should average at least the configured rate when measured over any time interval equal to or longer than the time it takes to send an output link MTU sized packet at the configured rate
    - The configured minimum rate MUST be settable by a network administrator
  - **Implementation of the EF PHB**
    - Simple Priority Queuing
    - Weighted Round Robin
    - Class Based Queuing
  - **Recommended Codepoint: 101110**

The Integrated Services Architecture

Integrated Services

- Extension of the classical Internet Architecture for QoS support (RFC 1633)
- The main IntServ characteristics are:
  - **Per-flow based**
  - **On-demand dynamic resource allocation**
  - **Relies on proper signaling protocols (e.g. RSVP)**
  - **Explicit Admission Control at each hop**
  - **Stateful approach (soft state)**

Service Classes

- Service Class:
  - **Specifies the type of service experienced by a flow**
  - **IntServ adds 2 service classes to the traditional IP Best Effort service:**
    - Guaranteed service
      - Suitable for applications with hard real-time requirements
    - Controlled load service
      - Developed for delay adaptive applications (with loose delay requirements)

Guaranteed Service

- Guaranteed Service (RFC 2212):
  - **Guaranteed Service Flows are characterized by:**
    - Assured level of bandwidth
    - Mathematically bounded end-to-end queuing delay
    - Absence of queuing loss
  - **The application involved specifies:**
    - Traffic characteristics $\Rightarrow$ Tspec = [r b p m M]
    - Reservation request $\Rightarrow$ Rspec = [R S]
  - **The Guaranteed Service flow is subject to:**
    - Traffic Policing:
      At network edges out-of-profile traffic is demoted to Best Effort
    - Traffic shaping:
      At all merge points traffic is reshaped to fit into Tspec

Controller Load Service

- Controlled Load Service (RFC 2211):
  - **Controlled Load Service Flows are characterized by the same service level of a Best Effort flow in an unloaded network**
  - **The application involved specifies only:**
    - Traffic characteristics ⇒ Tspec = [r b ~~p~~ m M]
  - **The Controlled Load flow is subject to Traffic Policing and Traffic Shaping in all network elements**

Delay Boundary (1/3)

- Theorem (Parekh and Gallager, 1993):

  Given an arbitrary network topology and under the following assumptions:

  - Flow characterized and enforced by a Tspec
  - WFQ queuing discipline used in all routers of the network
  - Reserved rate R not less than average Token Bucket rate ($R \geq r$)

  the maximum end-to-end queuing delay D_{QMax} is mathematically upper bounded

Delay Boundary (2/3)

- Maximum end-to-end delay is given by

$$D_{MAX} = D_{FIX} + D_{QMax}$$

  - **D_{FIX} is related to fixed delays (transmission, propagation)**
  - **D_{QMax} is the maximum end-to-end queuing delay**

- In the ideal case each flow would see a dedicated wire of bandwidth R:

$$D_{QMax} = b/R$$

Delay Boundary (3/3)

- Two error terms are introduced to take into account non idealities:

$$D_{QMax} = b/R + C/R + D$$

- Considering peak rate limitation *p* and maximum packet size *M*, D_{QMax} becomes:

$$D_{QMax} = \begin{cases} \dfrac{(b-M)(p-R)}{R(p-r)} + \dfrac{M+C_{TOT}}{R} + D_{TOT} & p > R \geq r \\ \dfrac{M+C_{TOT}}{R} + D_{TOT} & R \geq p \geq r \end{cases}$$

Guaranteed Service vs Controlled Load

	Guaranteed Service	Controlled Load Service
Typical applications	*Real time applications with tight delay requirements*	*Adaptive Real Time applications*
End-to-end behavior	*Bounded Delay* *Guaranteed Throughput* *No queuing losses*	*Approximates Best Effort over an unloaded network*
Parameters	*Tspec = [r b p m M]* *Rspec = [R S]*	*Tspec = [r b ~~p~~ m M]*
Exported information	*(C_{TOT},D_{TOT}) which measure deviation from ideal model*	*No exported information*

Resource ReSerVation Protocol

- RSVP (Resource ReSerVation Protocol – RFC 2205):
  - **Signaling protocol used to carry information about traffic characteristics and requested resources to all network elements involved in QoS support**
  - **Originally developed within the IntServ framework (RFC 2210) to let routers along the path to reserve resources**
  - **It can also be used as a signaling protocol in other scenarios (e.g. DiffServ)**

RSVP

- RSVP Design Principles:
  - **Receiver Initiated Reservation**
    - A reservation is set up, maintained and torn down by the receiver(s)
    - Each receiver decides the proper amount of resources to reserve based on its QoS needs/capabilities
    - Suitable for multicast reservations and heterogeneous receivers
  - **Soft State Approach**
    - Opposite to Hard State choice used in connection oriented networks
    - Path and Reservation states in routers along the path need periodic refreshes in order to avoid their expiration (time out)
    - Periodical refreshes are performed by hosts
    - This choice adapts to dynamic changes of routing paths

RSVP

- RSVP Design Principles (cont.):
  - **Independence from underlying routing protocol**
    - RSVP is NOT a routing protocol and relies on the underlying IP protocol to decide the routing path
    - This allows datagrams to be transparently treated in non-RSVP routers, even if QoS cannot be assured in this case
    - The path could include non-RSVP clouds
    - This choice was made in order to allow incremental deployment
  - **Extensibility**
    - RSVP messages carry opaque objects
    - Extension to the original protocol are possible by the definition of new objects

RSVP

- RSVP key concepts:
  - **Session**
    - Identifies the destination of a data flow
    - Defined by the triple:

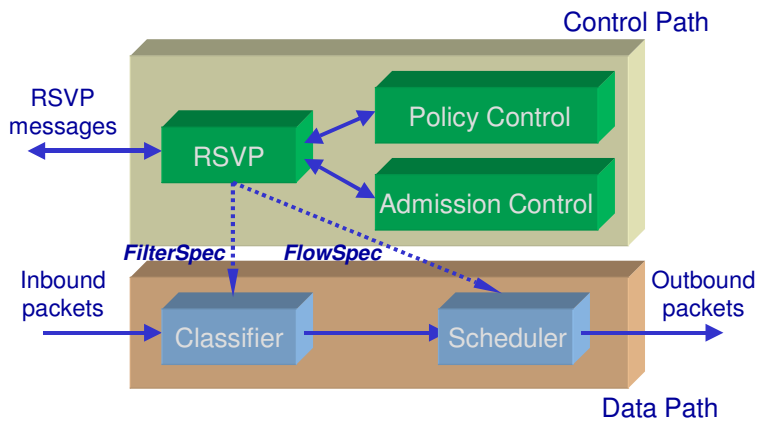
      (dest. IP address, dest. Port, protocol type)
  - **Flow Descriptor**
    - Contains information about a flow
    - It is composed of two parts:
      - FilterSpec
      - FlowSpec

RSVP

- Flow Descriptor (cont.):
  - **FilterSpec**
    - Identifies the packet subset entitled to receive QoS
    - Defined by the pair: (source IP address, source port)
    - Together with the session identifies a flow
    - Updates packet classifier
  - **FlowSpec**
    - Specifies both flow characteristics and QoS requests
    - Contains:
      - Service Class (Guaranteed/Controlled Load)
      - Traffic Characteristics (Tspec)
      - QoS Requests (Rspec)
    - Updates packet scheduler

IntServ Router

- An Integrated Service Router:



RSVP Messages

- RSVP messages are carried inside IP datagrams (protocol type 46)
- Each message refers to a Session and contains a Session Identifier
- There are 7 RSVP messages:
  - Path (downstream)
  - Resv (upstream)
  - Path Tear (downstream)
  - Resv Tear (upstream)
  - Path Err (upstream)
  - Resv Err (downstream)
  - Resv Confirm (downstream)

PATH

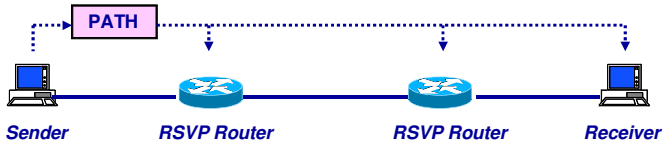
- Path message:
  - **Sent downstream towards receiver(s) in order to:**
    - Provide receivers with information about sender(s) Tspec
    - Determine the path followed by the flow and collect its characteristics
  - **Creates a Path State in all RSVP routers encountered**
  - **Contains the following objects:**
    - Session: (dest. IP address, dest. Port, protocol type)
    - Phop: IP address of the previous RSVP router
    - Sender Template: (sender IP address, sender IP port)
    - ADSPEC (Optional): One Pass With Advertisement (OPWA) information updated by routers along the path

RESV

- Resv message:
  - **Sent upstream towards sender(s)**
  - **Carries reservation requests to the routers along the multicast tree**
  - **Resv messages originated by receivers of the same multicast group are merged together before being forwarded upstream**
  - **Contains the following objects:**
    - Session: (dest. IP address, dest. Port, protocol type)
    - Flow Descriptor: (FilterSpec,FlowSpec)
    - Reservation Style: FF, SE or WF
    - ResvConf (Optional): IP address of the receiver that wants to be acknowledged for a successful reservation

Unicast Reservation

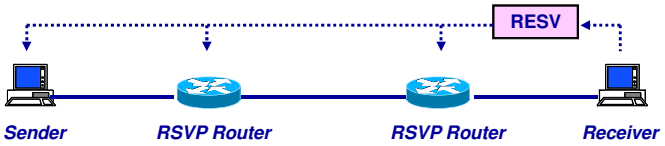
- Unicast Reservation: 1st step (Path message)



- Upon the reception of a Path message a RSVP router:
  - **Creates or updates a Path State associated to the corresponding session, containing *Phop*, *Sender_Tspec* and *Adspec***
  - **Restarts the cleanup timer associated to this Path State (refresh)**
  - **Updates *Phop* and *Adspec* objects and then forwards Path message towards next router (possibly a RSVP router)**

Unicast Reservation

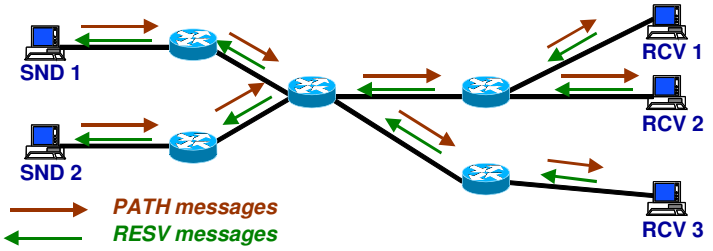
- Unicast Reservation: 2nd step (Resv message)



- Upon the reception of a Resv message a RSVP router:
  - **Analyzes FlowSpec for Admission Control**
  - **In case of success creates or updates a Resv State**
  - **Restarts the cleanup timer associated to this Path State (refresh)**
  - **If needed perform actual reservation of resources, updating parameters in the packet classifier and in the packet scheduler**
  - **Forwards Resv message to the previous RSVP node (Phop)**

Multicast Reservation

- Multicast Reservation:



  - **The procedure is stepped in two phases as before**
  - **Routers reserve resources on each outbound link by joining requests according to the reservation style**
  - **Reservations are merged before sending them upstream**

Integrated Services Problems

- The main problem of the IntServ architecture is scalability:
  - **Per-flow approach and soft-state philosophy are particularly suitable for multicast broadband applications**
  - **In point-to-point narrowband applications these choices imply large processing overhead in routers and great amount of traffic generation for periodic refreshes**
  - **Example:**
    ADPCM coding requires 32 kb/s for a voice channel. Neglecting packet overhead a single OC-12 interface of a backbone router (622 Mb/s) should support up to 20000 flows, implying that:
    - packet scheduler has to manage 20000 queues
    - up to 20000 states must be periodically refreshed