Integrated Services in the Internet

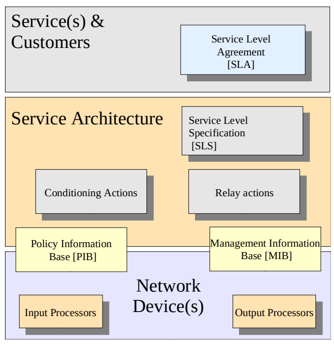
Lecture for S-38.180 QoS in the Internet

Mika Ilvesmäki

16.10.2001

The QoS story so far...

- Where are we in this lecture
  - Low level mechanisms (building blocks of the QoS) have been dealt with
    - Schedulers, queuing, routing
  - Time to advance to building service architectures using the building blocks
  - Time to apply engineering visions



Outline

- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

History

- It was 1991...
  - and there was not (that much) traffic in the internet
  - No WWW until 1993
  - no other multimedia... yet
    - multicast was already designed, but it was just starting with IETF audio- and videocasts
- However, some people anticipated problems due to multimedia-applications

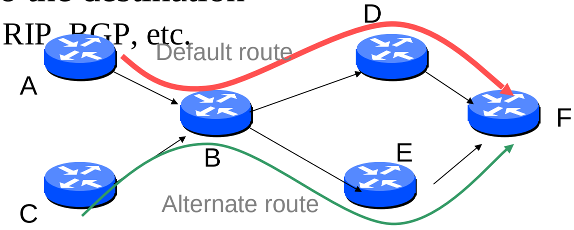
Application types

- Elastic
  - All tolerant ”old-fashioned” Internet applications
    - FTP, Usenet News, E-mail,
- Tolerant playback applications
  - One-way video feed, oneway broadcast
    - Some tolerance achieved with play-out buffers
- Intolerant playback applications
  - Applications that need data to be delivered in real-time
    - low delay, no jitters, enough bandwidth
  - Two way conversations (IP phone)

- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

Routing in the Internet

- Current Internet routing is based on finding the shortest path to the destination
  - No possibility to optimize resource usage
  - Destination based routing offers the possibility to use only the default route
- shortest path refers usually to the number of hops to the destination
  - OSPF, RIP, BGP, etc.



Domain wide QoS

- a.k.a Constraint based routing (CR) or QoS routing (QoSR)
- Calculate the route so that multiple constraints are met and that the route is optimal for every constraint
  - Constraints: delay, bandwidth, etc. and/or administrative
- Problems: route oscillation, path capacity
- Could be used together with a signalling protocol (RSVP) that has knowledge on the constraint values

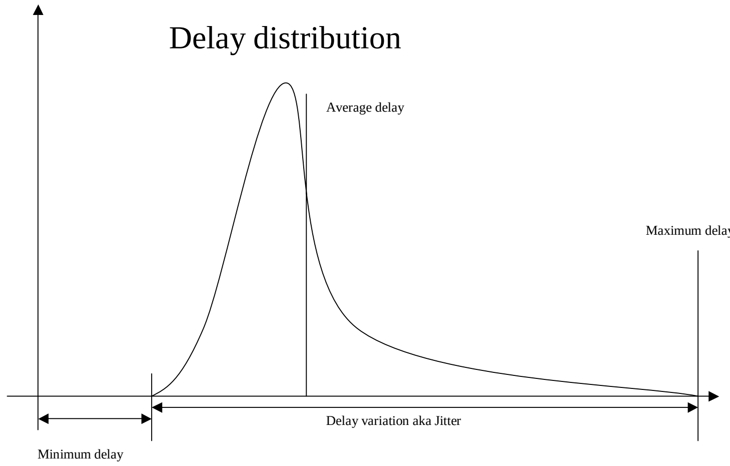
- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

Quantitative QoS-parameters

- *Available bit rate/ bandwidth*
  - How fast you are allowed to send packets to the network?
- *Packet discard / Data loss*
  - What packets are dropped in case of congestion?
- *Delay*
  - Time for the packet to reach its destination
  - How long is your data relevant?
- *Variation of delay / Jitter*
  - effectively kills the usability of Voice over IP –applications

- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

Delay and delay variation



- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

Original design objectives for IntServ

- Build a multicast network with videoconferencing ability
  - Only a few senders at a time
    - real-time
    - low packet loss
    - no congestion control (UDP)
  - VoIP not expected!!
- Protect multimedia traffic from TCP effects and vice cersa
- Objective: Preserve the datagram model of IP networks AND provide support for resource reservations and performance guarantees

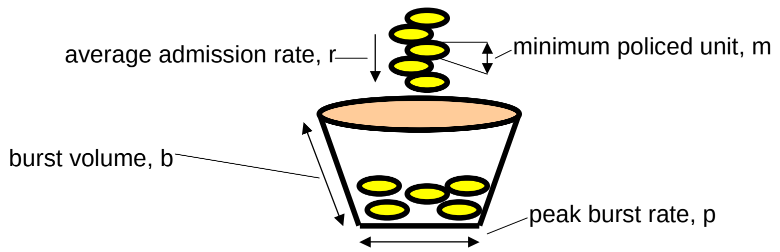
Integrated Services

- Provide Best Effort as before
  - no reservations for TCP traffic
  - possibility to use adaptive applications
  - sometimes BandWidth is enough
- Provide resources for multimedia traffic
  - multicast streams are long lasting, therefore state setups are ok
    - Caveat!!: VoIP is not OK !!
- Provide services for individual users and their applications!!
  - aka per-flow approach

- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

Flow model of IntServ

- Flow model described by a leaky bucket
  - token rate (r): 1 byte/s - 40 Terabytes/s
  - token-bucket depth (b): 1 byte - 250 Gbytes
  - peak traffic rate (p): 1 byte - 40 Terabytes/s
  - minimum policed unit (m): amount of data in the IP packet (other protocols, user data)
  - maximum packet size (M): maximum size of the packet within this flow (bytes)
    - larger packets do not receive the same QoS



Guaranteed service (RFC 2212)

- for non-adaptive applications requiring fixed delay bound and a bandwidth guarantee
- WFQ service (refer to lecture on queuing mechanisms)
- computes and controls the maximum queuing delay
- guarantees that packets will arrive within a certain delivery time and will not be discarded because of queue overflows, provided that flow's traffic stays within the bounds of its specified traffic parameters
- does not control minimal or average delay of traffic, nor is there control or minimization for jitter
- no packet fragmentation is allowed, packets larger than M are nonconforming.
- traffic policing with simple policing and reshaping

- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

Delay calculation for Guaranteed Service

End-to-end queuing delay:

$$Q_{delay} = \frac{(b-M)(p-R)}{R(p-r)} + \frac{(M+C_{tot})}{R} + D_{tot}, \text{ if } p > R \geq r$$

or:

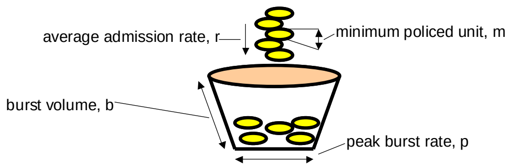
$$Q_{delay} = \frac{(M+C_{tot})}{R} + D_{tot}, \text{ if } R \geq p \geq r$$

- p=peak rate of flow (bytes/s)
- b=bucket depth (bytes)
- r=token bucket rate (bytes/s)
- R=bandwidth (service rate)
- m=minimum policed unit (bytes)
- M=maximum datagram size (bytes)
- C=packet delay caused by flow parameters (bytes)
- D=rate independent delay caused by network nodes

- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

TOKEN_BUCKET_TSPEC

- Guaranteed service is invoked by a sender specifying the flow parameters in the Tspec
- Controlled-load service is described in Tspec
- Describes traffic with
  - bucket rate (r)
  - peak rate (p)
  - bucket depth (b)
  - minimum policed unit (m) and maximum packet size (M)



- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

Rspec

- Receiver requests a desired service level with Rspec
- Used only in Guaranteed Service
- Describes the service requirements with
  - Service rate (R), R>=r, may be higher than requested
  - Slack Term (S), microseconds, describing the difference between the desired delay and the delay obtained by using a reservation level of R.

Controlled load service (RFC 2211)

- provides unloaded network conditions
  - for applications requiring reliable and enhanced best-effort service
  - aims to provide service that closely approximates traditional best-effort in a lightly loaded or unloaded network environment -> better than best effort
- intended for adaptive applications
  - applications provide network an estimation of the traffic it is about to send
  - acceptance (by the network) of a controlled load request implies a commitment to provide better than best-effort
- priority service with admission control
- no fragmentation, packets must comply to MTU

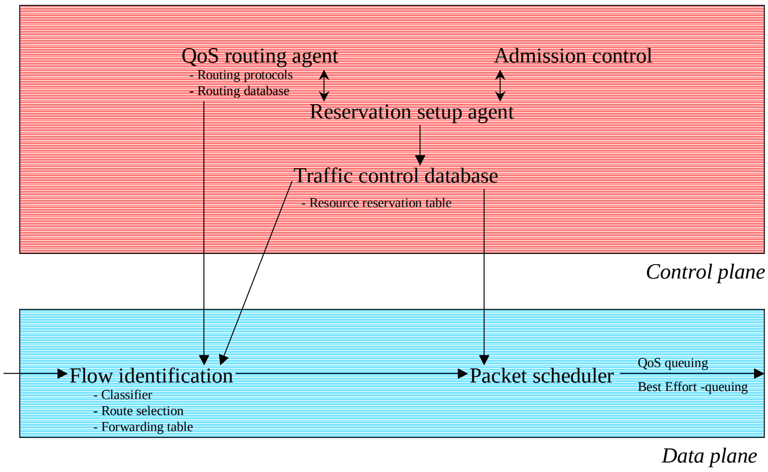
Your Internet connection request can not be completed at the moment. Please try again as soon as some resources are made available somewhere in the net.

QoS in the Internet-routers

- New router functionality
  - Traffic shaping
  - Admission control
    - To control resources
  - Differential congestion management
    - advanced queue management algorithms
    - CBQ, WFQ, etc.
  - Consistent handling of packets
    - State, ‘global’ knowledge of policy and QoS/CoS decisions

”There is an inescapable requirement for routers to be able to reserve resources in order to provide special QoS for specific user packet streams.”

IntServ router implementation reference model



- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

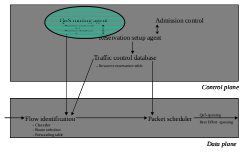
IntServ node characterization

- General descriptive parameters used to characterize the QoS capabilities of nodes in the path of a packet flow (RFC 2215)
  - NON_IS_HOP
    - the break bit indicating a break in the QoS chain
    - set by the device that is not IntServ compliant or knows such devices to exist in the path
  - NUMBER_OF_IS HOPS
  - AVAILABLE_BANDWIDTH
    - 1 byte/s ... 40 Terabytes/s
  - MINIMUM_PATH_LATENCY
    - speed-of-light + packet processing limitations
  - PATH_MTU
  - TOKEN_BUCKET_TSPEC
    - only used by the sender and the edge node

Router blocks: QoS routing

- Current Internet uses distributed route calculation
  - Every router decides for itself what is the best route to a given destination.
- In the future Internet route calculation has to be more centralized
  - Ability to compute the path at the source
  - Ability to distribute information about network topology and link attributes
  - Ability to do explicit routing
  - Resource reservations and link attrribute updates

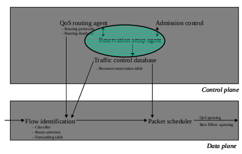
QoS routing is not specified in any manner within the IntServ!!



- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

Router blocks: Reservation setup

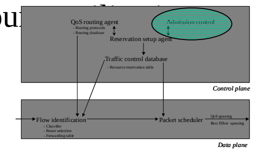
- Need for a protocol
  - RSVP
- Hop by hop state establishment
  - traffic characteristics
- Billing/accounting setup
- More on RSVP in the next lecture...



- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

Router blocks: Admission control

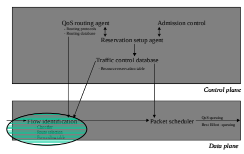
- Before a flow is accepted it has to pass the admission control test
- Parameter based
  - precise characterization of a traffic flow
  - difficulty of accurately modeling traffic
- Measurement based
  - probabilistic traffic characterization
  - good level of guarantee to resou ratio



- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

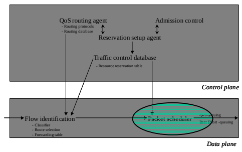
Router blocks: Flow identification

- Identify to what flows (if any) packets belong to
  - must be performed to every incoming packet
    - Multifield classification decides the appropriate queue
  - requires fast hardware if (and when) performed at wire speed
  - 64 byte packets arrive in 622 Mbit/s line back to back in less than 1μs



Router blocks: Packet scheduling

- Refer to the appropriate lecture on scheduling algorithms
  - WFQ
    - explained with the fluid model
  - GPS
  - PGPS
  - WF^{2}Q
  - Hierarchical WFQ
  - SCFQ, WRR, DRR, CRR etc. etc.



- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

Billing in Integrated Services

- In principle the billing could be arranged as in the POTS.
- In practise Internet routers and Internet in general has not been designed to collect and update the network usage of an individual user (scalability)

- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

Pricing/Billing alternatives

- Flat rate (even sum/month)
- Usage based
  - received data
  - sent data
  - use of resources (Bandwidth etc.)
- Billing based on user profile
  - Being a member of user group
  - Using certain applications (VoIP-phone vs. Web-browser)
- Combination of any and all of the above
- How complicated can a Internet-bill be so that the user may verify it and accept it?!

- History and preliminary concepts
  - types of Internet applications
  - general QoS concepts
- Concepts of IntServ
  - flow model
  - service classes
- Building the IntServ-router
  - routing, scheduling
  - Pricing/Billing basics
- Future notes

IntServ problems

- Resources
  - OK in small networks
    - provides for end-to-end exact QoS
  - What about large networks?
    - router capacity for resource reservation cannot be scaled on per-flow basis (in the Internet core)
- For IntServ to function, especially for Guaranteed Service, every node on the path must implement the IntServ functionality
- Router requirements are high
  - RSVP, admission control, MF classification and packet scheduling

The Future of Integrated Services?

- Millions of users are hard to manage one by one according to their individual wishes.
  - qualitative QoS -> not IntServ
- Scalability
  - If the amount of information grows faster or at the same pace in the core as it does at the edge the solution in question DOES NOT SCALE well.
- It is easier to decide *which* packet is forwarded and which dropped or delayed than to determine *when* a packet should be forwarded.
  - qualitative QoS -> not IntServ
- Qualitative is easier to implement than quantitative
  - IntServ is not likely to be the widely implemented QoS solution!!