Internet Technology

The "inner network" view, part 1: Quality of Service

Michael Welzl http://www.welzl.at

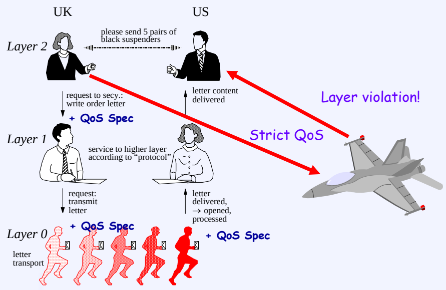
DPS NSG Team http://dps.uibk.ac.at/nsg
Institute of Computer Science
University of Innsbruck, Austria

Both sides of the story...

1. End users want:
   - Efficient Internet data transfer
     - e.g., 100 Mbit/s should really be 100 Mbit/s! (not always true! e.g. "theoretical" vs. "real" wireless bandwidth)
     - application (video, audio, ..) quality should be good
   - Cheap service
2. Service providers want:
   - Money!
     - Save money: efficient use of existing capacity
     - Earn extra money: provide special services with guarantees (e.g., video conferencing)

Thus, two major parts:
- **Efficient End-to-End Internet Data Transfer**
- **Quality of Service**

QoS and network layers



QoS and network layers /2

- QoS: fundamentally an end-to-end issue...
- QoS spec. must not be violated at any layer
- QoS request may originate from (almost) any layer
- QoS provisioning may be demanded at (almost) any layer
- There is no overall framework - demand for QoS often leads to layer violation

QoS below IP

- LAN: Medium Access Control (MAC) Layer
  - CSMA/CD (Ethernet): behaviour practically unpredictable (collisions lead to Binary Exponential Backoff, calculations too complicated)
  - Token passing schemes: bandwidth / delay predictable
- WAN: *ATM-Layer* (ATM has its own 3-dimensional model)
  - ATM was the first serious QoS attempt - *"ATM to the desktop"*
  - Constant cell size of (5+48) bytes enables **T**ime **D**ivision **M**ultiplexing -> predictable data rate!

ATM Services

CBR (Constant Bit Rate)	emulates a leased line
RT-VBR (Real-time Variable Bit Rate)	for rt-streams w/ varying bandwidth such as MPEG
NRT-VBR (Non-real-time Variable Bit Rate)	similar to RT-VBR, but more jitter is tolerated
ABR (Available Bit Rate)	Cheap service - you do what you are told, get what is available and achieve a small cell loss ratio
UBR (Unspecified Bit Rate)	Cheap, too: no promises - best used by IP
GFR (Guaranteed Frame Rate)	minimum rate guarantee + benefit from dynamically available additional bandwidth

ATM and reality

- ATM to the desktop: dead
  - A technology lesson! (L.2 complexity, QoS through layers, ..)
  - ATM: bad word in the IETF...
- Nowadays, most often used for high-speed IP links (backbone)
- Suboptimal for various reasons:
  - Cell size does not match packet sizes
  - IP provides datagram service, no use for CBR etc. (IP hourglass!)
  - IP mostly used with UBR or ABR service; in case of ABR, TCP is a control loop on top of a control loop!
  - Just too complex!

e.g., ACONET switched (almost completely) from ATM to Gigabit Ethernet in 2001 !

QoS in WiMAX (802.16)

- Connection oriented
  - QoS per connection; all services applied to connections
  - managed by mapping connections to "service flows"
  - bandwidth requested via signaling
- Three management connections per direction, per station
  - basic connection: short, time-critical MAC / RLC messages
  - primary management connection: longer, delay-tolerant messages authentication, connection setup
  - secondary management connection: e.g. DHCP, SNMP
- Transport connections
  - unidirectional; different parameters per direction
- Convergence sublayers map connections to upper technology
  - thus, also QoS!
  - two sublayers defined: ATM and "packet" (Ethernet, VLAN, IP, ..)

802.16 services

- Services designed for ATM compatibility
- Uplink scheduling types
  - Unsolicited Grant Service (UGS)
    - for real-time flows, periodic fixed size packets
    - e.g. VoIP or ATM CBR
  - Real-Time Polling Service (rtPS)
    - for real-time service flows, periodic variable size data packets
    - e.g. MPEG
  - Non-Real-Time Polling Service (nrtPS)
    - for non real-time service flows with regular variable size bursts
    - e.g. FTP or ATM GFR
  - Best Effort (BE)
    - for best effort traffic
    - e.g. UDP or ATM UBR
- Specified via QoS parameters
  - max. sustained traffic rate / traffic burst, min. reserved traffic rate
  - vendor specific parameters

Typical QoS requests

ATM	Peak / Sustained / Minimum Cell Rate, Cell Delay Variation Tolerance, Cell Transfer Delay, Cell Error Rate, Cell Loss Ratio, ..
Layer 4 (distributed Multimedia app)	Throughput, End2end Delay, Residual Error Rate (not (yet?) on the Internet!), Connection Establishment Delay / Failure Probability, ..
Layer 7	Transmission Security, Data Encoding Completeness, ..
Human Layer	Perceived quality - "does it look good?", "does it feel controllable?", fun factor, ..

QoS Architectures

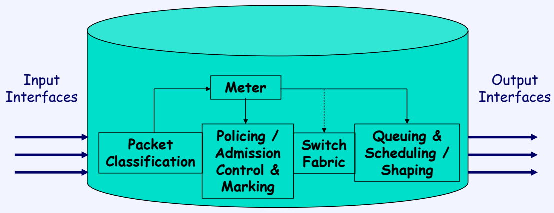
- Only of historical value
- Heidelberg QoS Model, OMEGA, int-serv, XRM (hierarchical), QoS-A and Tenet (3-dimensional), OSI, TINA, MASI, ..
- Various concepts: related to layers (OSI, QoS-A), related to specific implementations (int-serv), ..
- Architectures identify fundamental concepts of QoS specification, provisioning, control and management
- No overall agreement on a single architecture

IP QoS

- Interview with Van Jacobson, EE Times *http://www.eetimes.com/* "TCP/IP pioneer's past is prologue", 03/07/2005

  "From my point of view, ATM was a link-layer technology, and IP of course could run on top of a link layer, but the circuit-oriented developers had interpreted the link layer as the network. The wires are not the network."
- IP = binding element across link layer technologies
  - Everything over IP, IP over everything!
- "ATM to the Desktop" failed - so, do it with IP

Generic QoS-capable router



Building blocks of modern QoS architectures

QoS router building blocks

- Packet Classification
  - Group packets according to header properties
  - Multiple fields (MF classification) needed to detect individual flows: ip source / destination, protocol and port numbers problems: packet fragmentation (port numbers), header compression, encryption (IPSec)
- Meter
  - Monitor traffic characteristics (e.g., does flow 741 hold its promises?), provide information to other block(s)
- Policing
  - Drop packets if certain conditions are fulfilled
- Admission Control
  - React (not necessarily drop packets) if certain conditions are fulfilled
- Marking
  - Mark packets (change header) if certain conditions are fulfilled
  - for later special treatment - maybe not even in the same router

QoS router building blocks /2

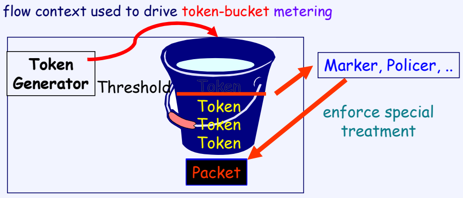
- Switch(ing) Fabric
  - Do a query on the routing table, decide where to send the packet
- Queuing
  - If a packet cannot be delivered immediately (congestion), put in queue(s) for later delivery
  - Decision: which queue? Active queue management?
- Scheduling
  - When to take a packet from which queue (e.g., round robin)
- Shaping
  - Adjust traffic characteristics if certain conditions are fulfilled (usually implemented in scheduling)
  - Useful even without QoS provisioning: Do not exceed max. promised quality - customers will get accustomed and complain!

Integrated Services (IntServ)

- Notion:
  hard guarantees desired, per-flow resource reservation needed
- Two services defined:
  - Guaranteed Service
    guaranteed bandwidth, firm bounds on end-to-end queuing delays; to be used by real-time applications
  - Controlled Load
    closely approximates the behaviour seen when there is (almost) no congestion; to be used by elastic applications
- Architecture, Services / Reservation signaling protocol ("Resource Reservation Protocol" - RSVP) design separated

IntServ per-hop requirements

- Classification:
  - per-flow context established via multifield classification
  - flow context used to drive token-bucket metering

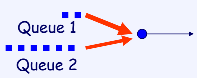


  - implemented as byte counter; goal: detect various degrees of burstiness
  - several thresholds (also: empty) with associated treatment possible!

IntServ traffic specification contains token generation rate, bucket size

IntServ per-hop requirements /2

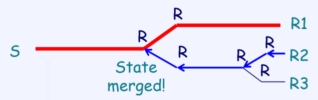
- IntServ token bucket metering leads to remarking or dropping (admission control)
- Multiple queues, one for each flow
- Implementation: virtual queues - only one real queue per service
- Scheduler takes packets based on priorities *(airline analogy)*
  - e.g., 1, 1, 2, 1, 1, 2, .. but not priority queuing (q1 until empty) - may cause starvation of q2!



- No bandwidth guarantees because of packet sizes!
- Solution: Weighted Fair Queuing (WFQ), Class Based Queuing (CBQ)

Resource ReserVation Protocol (RSVP)

- Signaling - routers must know which flows to choose
- state in routers is established via PATH messages from sender
- Sender advertises allowed traffic spec via adspec messages
- Receivers initiate reservation (resv messages containing flow spec.)
- Multicast support, state merging:

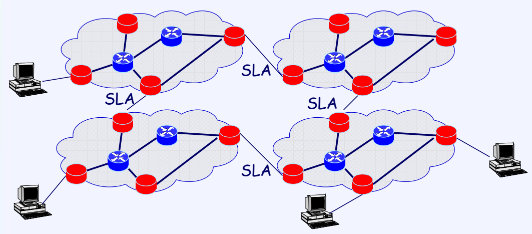


IntServ / RSVP discussion

- RSVP requires support by all routers
  (if unsupported, RSVP is tunneled - but no more hard guarantees)
- Scaling: per-flow state not feasible!
  RSVP protocol not scalable either (maybe due to bad implementation)
- Strict guarantees per customer: complicated accounting
- Solution: "softer" QoS, no per-flow state in core routers - DiffServ

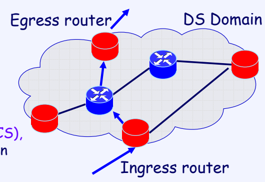
	Best-Effort	IntServ/RSVP	DiffServ
QoS-Guarantees	none	flow-based	aggregated
Configuration	none	dynamic end2end	static edge2edge
Scalability	100%	limited	more

Differentiated Services (DiffServ)



- Edge routers: Classifier / Meter / Marker / Shaper / Dropper
- Core routers: static forwarding according to DiffServ-class, implementation may vary
- SLA: Service Level Agreement between DS Domains

DiffServ terminology



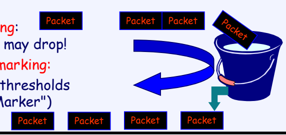
- SLA contains non-technical aspects
- Service Level Specification (SLS):
  - Parameters which determine the service provided by a DS domain
  - contains Traffic Conditioning Spec. (TCS), and other properties such as encryption and routing constraints
- DiffServ Codepoint (DSCP) - IPv4 Precedence / TOS Bytes
- DSCP mapped to Per Hop Behaviour (PHB)
  - how are packets treated in the core?
  - Aggregated flows with same DSCP: Behaviour Aggregate (BA)
  - Distinguish: PHB specification / implementation
  - PHB Group: PHBs that call for similar spec. / implementation

DiffServ details

- Edge routers: MF and BA classification
  based on signaling, metering .. or ideas such as simply UDP / TCP
- Expedited Forwarding (EF) PHB
  - "Virtual Leased Line" Service
  - Aggregated flows must not exceed peak bandwidth
  - Ingress Router: Policing (dropping); Egress Router: shaping
  - Small delay - real time apps; simple service model
  - Unused bandwidth used by best-effort traffic!
- Assured Forwarding (AF) PHB Group
  - Supports bursty flows
  - Packets are marked with AF Class and Drop Precedence
  - non-conforming packets are remarked

DiffServ details /2

- DiffServ does not define:
  - End2end service models
  - Implementation details (PHBs, traffic conditioners, ..)
- But: hints
- As in ATM ABR, "open" spec. leads to a lot of research work
- Implementation examples:
  - schedulers for PHB: WFQ, CBQ, WRR (Weighted Round Robin), ..
  - policers for drop precedence: Weighted RED, RIO - RED variants which drop according to priorities
  - shapers for traffic conditioning:
    Leaky Bucket - enforces CBR, may drop!
  - meters for drop precedence marking:
  - Token Bucket(s) with various thresholds ("A Single Rate Three Color Marker")



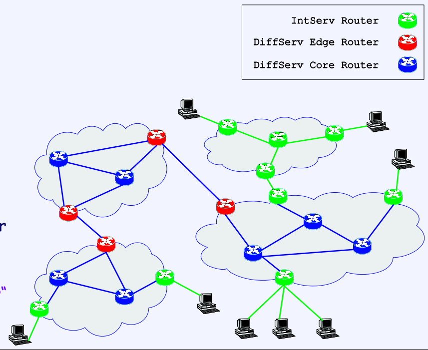
DiffServ extensions / ideas

- IntServ over DiffServ
  - may be good idea: fine granularity of IntServ / RSVP signaling at edge routers & end systems, scalability through DiffServ core
  - IntServ flows are aggregated for DiffServ
  - DiffServ does not participate in RSVP signaling
  - IntServ treats DS Domains (EF PHB!) as a leased line
- Bandwidth Broker
  - additional network nodes for signaling and negotiation
  - translation: SLS -> TCS
  - explicit communication with edge routers, e.g. via COPS
- Open specification brought some chaos, too:
  Red / green / blue packets, assured / premium service, Gold / Silver / Bronze = olympic services .. **what is real?**

IntServ over DiffServ

Flexibility, service granularity of
IntServ
+
Scalability of
DiffServ
+
(sometimes) separate entity for QoS negotiation:

"bandwidth broker"



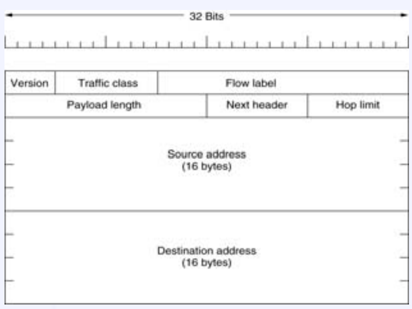
QoS Routing

- IntServ and DiffServ assume shortest-path routing!
  Not always optimal; some flows may prefer a *"long, fat pipe"*
- Solution: classify / meter, then forward according to requirements
- Knowledge of a path's QoS properties: additional routing metrics (increases routing protocol traffic!)
- Problems:
  - scalability / oscillation - if QoS Routing is done for many sources: quality reduced by own payload! use old path again?
  - when / how often is QoS measured / calculated?
- QoS Routing not yet a real issue in IETF
  (WG only produced framework, OSPF QoS extensions experimental)

Internet Protocol Version 6 (IPv6, IPNG)

- Different addresses (much bigger! but makes migration hard)
- Some header fields removed
- Multicast - IGMP now part of ICMP
- Mobility
- New optional header extensions
  (IPSec problematic for MF classification!)
- QoS support:
  - DiffServ field / flow label instead of ToS / precedence
    ...for easier flow classification (no further semantics defined)

Main IPv6 Header



Fragmentation: only in hosts!

Optional: extension headers

IP QoS lessons learned

Some QoS rules

- Scalability above everything!
  - especially avoid per flow state
  - avoid state alltogether
  - consider hierarchical structures for state aggregation
- QoS guarantees need a consistent end2end service model
- If hard guarantees are impossible, consider "softer" QoS
- Consider interactions with end system congestion control!
- Layer violations may be necessary
- Either "manage unfairness" or be fair (Internet: TCP-friendly)

History

- 1968 ARPAnet effort startet by BBN
- 1969 first protocols developed
- 1986 congestion collapse
- 1988 "Congestion Avoidance and Control"
- 1989? QoS discussions in the IRTF
- 1993 "Random Early Detection Gateways for Congestion Avoidance"
- 1994 IETF WGs on IntServ and RSVP
- 1995 IPv6
- 1998 RFC on Active Queue Management
- 1998 IETF WG on DiffServ
- 1999 RFC on Explicit Congestion Notification
- 2000 RFC 2990

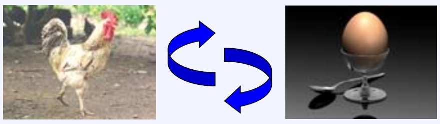
RFC2990 (IAB) - open issues

- State and Stateless QoS:
  IntServ & DiffServ are endpoints of a continuum of control models
- Uncertain: QoS-enabled applications or just transport layer?
  Each approach has its own advantages / disadvantages
- IntServ: explicit signaling - but DiffServ?
- Signaling of resource availability in the network core:
  DiffServ lacks signaling, IntServ/RSVP too fine-granular
- Still no standardized Inter-Domain signaling

RFC2990 (IAB) - open issues /2

- Trouble with TCP
  bursty by nature (ACK-clocking problem mentioned in RFC)
  token bucket = TCP-hostile
  should be managed in TCP stack
- Missing QoS routing / resource management solution
  IntServ and DiffServ assume regular shortest-path routing!
  Not feasible - traffic should be split accordingly
- QoS Accounting is still not solved
- Chicken (admins waiting for apps) /
  Egg (app developers waiting for admins) problem

QoS as an end user service



ISP:
- wants to max. revenue
- Install QoS alone: -$
- Provide QoS: ++$
  ...iff applications use it!

App developer:
- wants to max. revenue
- Implement QoS support: -$
- Support QoS: ++$
  ...iff ISPs provide it!

- Resembles prisoner's dilemma
- Can be solved with coordination (e.g. flow of $$$)
- How to coordinate apps + all ISPs along the path?

RFC2990 (IAB) - open issues /3

- Still no clear objectives
  application-centric vs. network-centric goals
- Unresolved security issues
  weighted fairness needs contract
- End-to-end architecture is needed
  Customers want QoS across the Internet
- **"It is extremely improbable that any single form of service differentiation technology will be rolled out across the Internet and across all enterprise networks."**

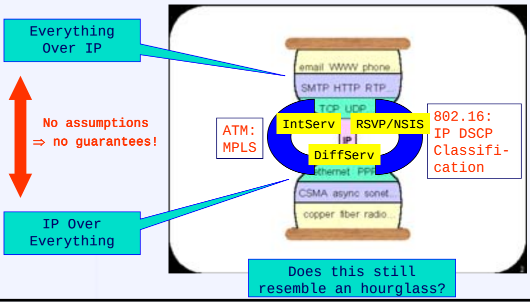
RFC2990 (IAB) - actual prediction

- "The architectural direction that appears to offer the most promising outcome for QoS is not one of universal adoption of a single architecture, but instead use a tailored approach where scalability is a major design objective and use of per-flow service elements at the edge of the network where accuracy of the service response is a sustainable outcome."
- "Architecturally, this points to no single QoS architecture, but rather to a set of QoS mechanisms and a number of ways these mechanisms can be configured to ineroperate in a stable and consistent fashion."

Further issues

- Heterogeneous environments (convergence = big issue!)
  - Problems with TCP over wireless links
  - Interactions with new underlying technologies (GPRS, UMTS, ..)
  - Problems with TCP over satellite links
- Will TCP still be a good match, anyway?
  - Congestion Control over "leased line"
- Security
  - Today, we have all got best effort.
  - Tomorrow, you may want to steal my service!
  - DoS (Degradation-of-Service) attacks?

Technology may no longer be the problem!



Charging, billing & accounting

- Most tools are there ... but:
- No significant progress in global standardization of charging, billing & accounting areas
- Numerous complicated research efforts to calculate prices based on QoS, but the IETF is behind
- Good global set of regulations needed (how much is given to which domain admins so they can add more bandwidth? What about inter-domain links?, ..) - may be the most difficult part
- **analogy: still no global laws for the Internet!**

The state-of-the-art

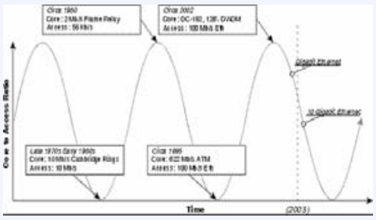
Papers from SIGCOMM'03 RIPQOS Workshop: "Why do we care, what have we learned?"

- QoS`s Downfall: At the bottom, or not at all! Jon Crowcroft, Steven Hand, Richard Mortier,Timothy Roscoe, Andrew Warfield
- Failure to Thrive: QoS and the Culture of Operational Networking Gregory Bell
- Beyond Technology: The Missing Pieces for QoS Success Carlos Macian, Lars Burgstahler, Wolfgang Payer, Sascha Junghans, Christian Hauser, Juergen Jaehnert
- Deployment Experience with Differentiated Services Bruce Davie
- Quality of Service and Denial of Service Stanislav Shalunov, Benjamin Teitelbaum
- Networked games --- a QoS-sensitive application for QoS-insensitive users? Tristan Henderson, Saleem Bhatti
- What QoS Research Hasn`t Understood About Risk Ben Teitelbaum, Stanislav Shalunov
- Internet Service Differentiation using Transport Options:the case for policy-aware congestion control Panos Gevros

Practical use of QoS

- Nowadays, IntServ, RSVP, DiffServ, ... are traffic management tools!
- Separation / routing of traffic based on characteristics
  - requires classification
  - may require metering
  - may require shaping
- Example: protect TCP from "greedy" UDP traffic
- Example: use different queues for file downloads and VoIP
- Note: overprovisioning = attractive alternative
  - manpower = expensive
  - But not always feasible (e.g. wireless networks)

Evolution of access vs. core bandwidth



Source: Jon Crowcroft's RIPQoS talk, http://www.cl.cam.ac.uk/~jac22/talks/ripqos.htm

- It seems that we're moving towards a shift...
  - May already have happened in some parts of the world (e.g. Japan)

- Will overprovisioning become too expensive?
  - Did QoS mechanisms simply appear at the wrong time?

Current QoS-related IETF activities

- Pre-Congestion Notification (PCN)
  - "The Congestion and Pre-Congestion Notification (PCN) working group develops mechanisms to protect the quality-of-service of established inelastic flows within a DiffServ domain when congestion is imminent or existing. These mechanisms operate at the domain boundary, based on aggregated congestion and pre-congestion information from within the domain."
  - Admission control, using the ECN field (which is not to be used for "normal" ECN within a PCN-cloud)

- Fairness work by Bob Briscoe
  - Rebuttal of "flow rate fairness" as a reasonable fairness measure
  - re-Feedback, specifically re-ECN: technical solution towards making users accountable for being unfair, using traffic shaping
  - IETF future unsure, conflict with ECN Nonce about usage of ECT(1)

QoS and the Grid

- Required participation of end users and all intermediate ISPs
  - "normal" Internet users want Internet-wide QoS, or no QoS at all
  - In a Grid, a "virtual team" wants QoS between its nodes
  - Members of the team share the same ISPs - flow of $$$ is possible

- Technical inability to provision individual (per-flow) QoS
  - "normal" Internet users
    - unlimited number of flows come and go at any time
    - heterogeneous traffic mix
  - Grid users
    - number of members in a "virtual team" may be limited
    - clear distinction between bulk data transfer and SOAP messages
    - appearance of flows mostly controlled by machines, not humans

- ⇒ QoS can work for the Grid !

References

Recommended reading

- Grenville Armitage, "Quality of Service in IP Networks", MTP (Macmillan Technical Publishing), USA, April 2000

  G. Huston, "Next Steps for the IP QoS Architecture", RFC 2990, November 2000

  Torsten Braun, "Internet Protocols for Multimedia Communications", IEEE Multimedia July-September 1997 (pt. 1) and October-December 1997 (pt. 2)

- Christina Aurrecoechea, Andrew T. Campbell and Linda Hauw, "A Survey of QoS Architectures", ACM / Springer Verlag Multimedia Systems Journal, Special Issue on QoS Architecture, Vol. 6 No. 3, pg 138-151, May 1998