QoS services and application-level service interfaces

IP “service”

- IP datagram service:
  - datagrams are subject to loss, delay, jitter, mis-ordering
- Performance: no guarantees
- **Integrated Services:**
  - new QoS service-levels
- **Differentiated Services:**
  - class of service (CoS)
- User/application may need to signal network
- User/application may need to signal other parts of application

Questions

- Can we do better than **best-effort**?
- What support do real-time flows need in the network?
- What support can we provide in the network?
- QoS for many-to-many communication?
- Application-level interfaces?
- Signalling

INTSERV

Questions

- What support do we need form the network to give QoS capability to the Transport layer?
- How can we control congestion in the network?
- How can we support legacy network protocols over the Internet?

Integrated services

- Need:
  1. service-levels
  2. service interface – signalling protocol
  3. admission control
  4. scheduling and queue management in routers
- Scenario:
  - application defines service-level
  - tells network using signalling
  - network applies admission control, checks if reservation is possible
  - routers allocate and control resource in order to honour request

INTSERV

- http://www.ietf.org/html.charters/intserv-charter.html
- Requirements for Integrated Services based on IP
- QoS service-levels:
  - current service: **best-effort**
  - **controlled-load service** (RFC2211)
  - **guaranteed service** (RFC2212)
  - other services possible (RFC2215, RFC2216)
- Signalling protocol:
  - RSVP (RFC2205, RFC2210)

INTSERV service templates

- Describe service semantics
- Specifies how packets with a given service should be treated by network elements along the path
- General set of parameters
  - <service_name>.<parameter_name>
  - both in the range [1, 254]
- TSpec: allowed traffic pattern
- RSpec: service request specification

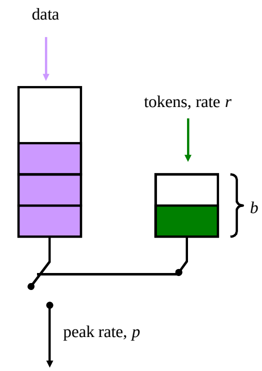
Some INTSERV definitions

- Token bucket (rate, bucket-size):
  - token bucket filter: total data sent $\leq$ (r*t* + b)
- Admission control:
  - check before allowing a new reservation
- Policing:
  - check TSpec is adhered to
  - packet handling may change if TSpec violated (e.g. degrade service-level, drop, mark, etc.)
- Characterisation parameters: local and composed

Token bucket (recap)

Token bucket

- Three parameters:
  - *b*: bucket size [B]
  - *r*: bucket rate [B/s or b/s]
  - *p*: peak rate [B/s or b/s]
- Bucket fills with tokens at rate *r*, starts full
- Tokens allow transmission
- Burst allowed at rate *p:*
  - data sent < *rt* + *b*
- (Also *m* and *M*)



General INTSERV parameters

- NON_IS_HOP (flag): no QoS support
- NUMBER_OF_IS_HOPS: QoS-aware hop count
- AVAILABLE_PATH_BANDWIDTH
- MINIMUM_PATH_LATENCY
- PATH_MTU
- TOKEN_BUCKET_TSPEC:
  - r (rate), b (bucket size), p (peak rate)
    m (minimum policed unit), M (maximum packet size)

Controlled-load service

- Best-effort **under unloaded conditions**:
  - probabilistic guarantee
- Invocation parameters:
  - TSpec: TOKEN_BUCKET_TSPEC
  - RSpec: none
- Admission control:
  - Class-Based Queuing (CBQ), priority and best-effort
- Policing:
  - not defined (e.g. treat as best-effort)

Guaranteed service [1]

- **Assured data rate with bounded-delay**
  - deterministic guarantee
  - no guarantees on jitter
- Invocation parameters:
  - TSpec: TOKEN_BUCKET_TSPEC
  - RSpec: R (rate), S (delay slack term, µs)
- Admission control:
  - Weighted Fair Queuing (WFQ)
- Policing:
  - drop, degrade to best-effort, reshape (delay)

Guaranteed Service [2]

- End-to-end delay bound:
  - maximum delay
  - based on fluid flow model
  - fluid flow model needs error terms for IP packets
- Error terms:
  - each router holds C and D
- C [B]: packet serialisation
- D [µs]: transmission *through* node
- Composed values:
  - C_{SUM} and D_{SUM}

$$delay = \frac{(b-M)(p-R)}{R(p-r)} + \frac{(M + C_{SUM})}{R} + D_{SUM} \qquad p > R \geq r$$

$$delay = \frac{(M + C_{SUM})}{R} + D_{SUM} \qquad R \geq p \geq r$$

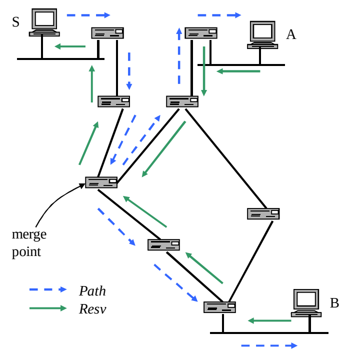
RSVP

INTSERV: RSVP [1]

- Provides signalling:
  - user-to-network
  - network-to-network
- Traffic information – *FlowSpec*:
  - *TSpec*
  - sent through network
  - *AdSpec* (optional)
- Receiver confirms reservation:
  - uni-directional reservation

INTSERV: RSVP [2]

- Two-pass, with soft-state:
  - sender: *Path* message
    - NEs hold **soft-state** until *Resv*, *PathTear* or time-out
  - receiver(s): *Resv* message - TSpec (+RSpec)
  - sender: *PathTear*
  - receiver(s): *ResvTear*
  - soft-state refreshed using *Path* and *Resv*
- Composed QoS params:
  - *AdSpec* for path



DigiComm II-17

Reservation types and merging

- *FilterSpec*: style of reservation
- Fixed-filter (FF):
  - *FilterSpec* required
  - distinct sender reservation
  - explicit sender selection
- Wildcard-filter (WF):
  - *FilterSpec* not required
  - shared sender reservation
  - wildcard sender selection
- Shared-explicit (SE):
  - *FilterSpec* required
  - shared sender reservation
  - explicit sender selection
- Merging reservation info:
  - merging allows aggregation of reservation information
  - merging not possible across styles
  - merging possible for reservations of the same style – use maximum

DigiComm II-18

Reservations about reservations

- Two-pass – one reservation may “block” another:
  - *PathErr* and *ResvErr*
- Need to hold a lot of soft-state for each receiver
- Extra traffic due to soft-state refreshes
- Heterogeneity limitations:
  - same service-level
- Router failure:
  - QoS degrades to best-effort, need to re-negotiate QoS
- Applications and routers need to be RSVP aware:
  - legacy applications
- Charging

DigiComm II-19

DIFFSERV

DigiComm II-20

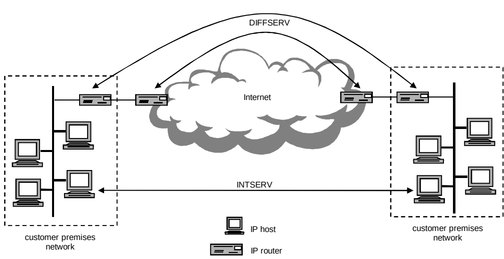
DIFFSERV

- http://www.ietf.org/html.charters/diffserv-charter.html
- Differentiated services:
  - tiered service-levels
  - service model (RFC2475)
  - simple packet markings (RFC2474)
- Packets marked by network, not by application:
  - will support legacy applications
- Simpler to implement than INTSERV:
  - can be introduced onto current networks

Service Level Agreements

- Not (necessarily) per-flow:
  - aggregate treatment of packets from a "source"
- Service classes:
  - Premium (low delay) - EF (RFC2598)
  - Assured (high data rate, low loss) - AF (RFC2597)
- **Service level agreement (SLA)**:
  - **service level specification (SLS)**
  - policy between user and provider - policing at ingress
  - service provided by network (end-system unaware)

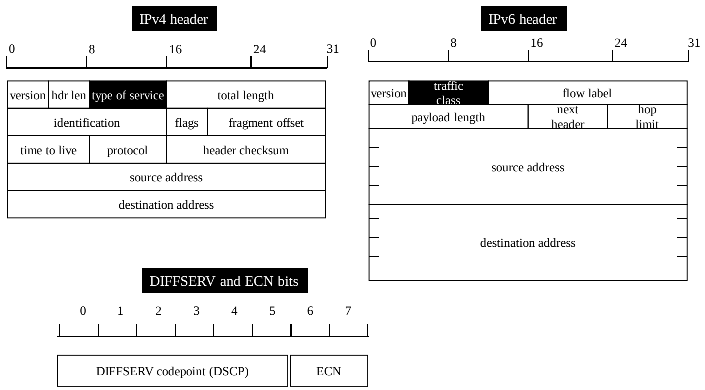
Scope of DIFFSERV



DIFFSERV classification [1]

- Packet marking:
  - IPv4 ToS byte or IPv6 traffic-class byte
  - **DS byte**
- Traffic classifiers:
  - **multi-field (MF)**: DS byte + other header fields
  - **behaviour aggregate (BA)**: DS field only
  - **DS codepoint**: values for the DS byte
- Aggregate per-hop behaviour (PHB):
  - aggregate treatment within network

DIFFSERV classification [2]

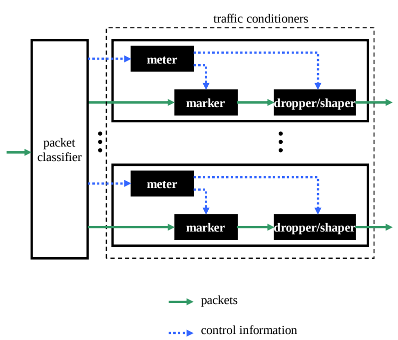


DIFFSERV PHBs

- Specify rate/delay in SLS
- **Expedited Forwarding (EF)** (RFC2598):
  - virtual leased line (VLL) service
  - data rate specified in SLS
  - low delay, low jitter, low loss
- **Assured Forwarding (AF)** (RFC2597):
  - 4 classes (1-4)
  - 3 levels of drop precedence per class (1-3)
  - AF11 - “best”, AF43 - “worst”

DIFFSERV traffic conditioning

- Traffic conditioners:
  - meter
  - marker
  - shaper/dropper
- Metering of traffic:
  - in-profile
  - out-of profile
- Re-marking:
  - new DS codepoint
- Shape/drop packets



DIFFSERV service invocation

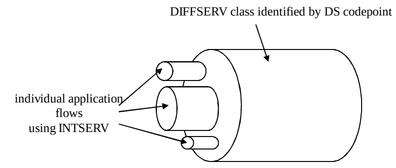
- At subscription:
  - per user/user-group/site/customer
  - multi-field, policy-based
- Within organisation:
  - per application/user/user-group
  - use ad hoc tools or network management system
  - behaviour aggregate or multi-field possible
- Dynamically using RSVP: IETF work in progress

Problems with DIFFSERV

- No standard for SLAs:
  - same DS codepoints could be used for different services by different providers
  - different providers using the same PHBs may have different behaviour
  - need end-to-end/edge-to-edge semantics
- Lack of symmetry:
  - protocols such as TCP (ideally) require symmetric QoS
- Multicast:
  - support for multi-party, symmetric communication?

INTSERV and DIFFSERV [1]

- Complimentary:
  - DIFFSERV: aggregate, per customer/user/user-group/application
  - INTSERV: per flow
- For example:
  - INTSERV reservations within DIFFSERV flows (work in progress)



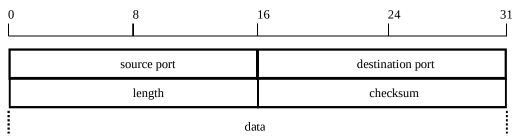
INTSERV and DIFFSERV [2]

	INTSERV	DIFFSERV
signalling	from application	network management, application
granularity	flow	flow, source, site (aggregate flows)
mechanism	destination address, protocol and port number	packet class (other mechanisms possible)
scope	end-to-end	between networks, end-to-end

RTP

UDP

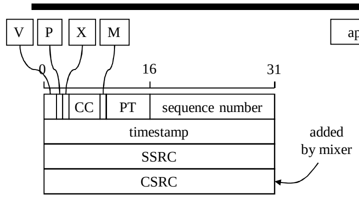
- Connectionless, unreliable, unordered, datagram service
- No error control
- No flow control
- No congestion control
- Port numbers
- Must be used for real-time data:
  - TCP automatic congestion control and flow control behaviour is unsuitable



RTP

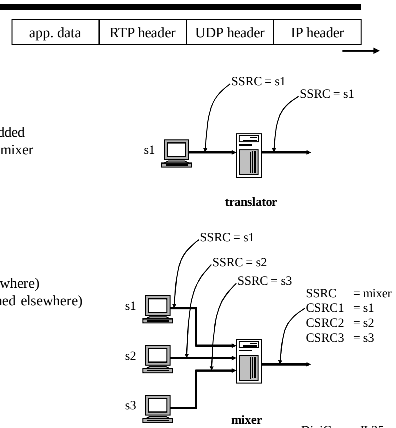
- RFC1889: general message format
  - specific formats for media types in other RFCs
- Carried in UDP packets:
  - application must implement reliability (if required)
  - supports multicast and point-to-point
- RTCP - Real Time Control Protocol:
  - application-level information (simple signalling)
- **RTP and RTCP provide no QoS guarantees:**
  - QoS mechanisms are separate

RTP header information



V 2-bits, version number (=2)
P 1-bit, indicates padding
X 1-bit, indicates extension header present
CC 4-bits, number of CSRCs (CSRC count)
M 1-bit, profile specific marker (defined elsewhere)
PT 7-bits, payload type, profile specific (defined elsewhere)
SSRC synchronisation source
CSRC contributing source

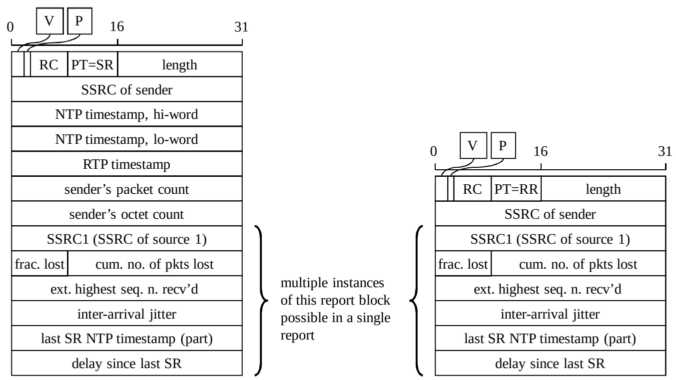
timestamp has profile/flow-specific units



RTCP - Real time Control Protocol

- Provides feedback to senders/receivers
- QoS info for flow:
  - packet info: loss, delay, jitter
  - end-system info: user info
  - application-specific or flow-specific info
- RTCP message types:
  - RR and SR: Receiver Report and Sender Report
  - SDES: Source DEScription
  - BYE: leave a RTP session
  - APP: application-specific

SR and RR messages



SDES

- Source DEScription: all ASCII strings
- Information types from RFC1889:
  - CNAME: canonical identifier (mandatory)
  - NAME: name of user
  - EMAIL: address user
  - PHONE: number for user
  - LOC: location of user, application specific
  - TOOL: name of application/tool
  - NOTE: transient messages from user
  - PRIV: application-specific/experimental use

BYE and APP

- BYE - leave RTP session:
  - SSRC (or SSRC and CSRC list if mixer)
  - reason for leaving
- APP - application-specific packets:
  - SSRC (or SSRC and CSRC list if mixer)
  - ASCII string for name of element
  - application-specific data

Application-level signalling

User-to-network

- Telco network:
  - common channel signalling (CCS)
  - separate data path and signalling path
  - equipment designed to handle data and signalling separate
- IP:
  - RSVP carried in IP packets along data path
  - scaling issues (RFC2208)
  - need aggregated signalling towards the core (use INTSERV with DIFFSERV?)

User-to-user signalling

- Call/session set-up
- Capabilities exchange
- Directory services
- PBX-like facilities
- Application-level signalling supported by network
- MMUSIC IETF WG:
  - application architecture
  - SDP
  - SIP (now has its own WG)
- H.323:
  - umbrella document for existing standards
  - uses ITU and IETF standards
  - currently more mature than MMUSIC work
  - wide support available (e.g. Microsoft NetMeeting)
  - IMTC: www.imtc.org

Summary

- Need QoS mechanisms for IP
- Per flow:
  - INTSERV
  - RSVP
  - does not scale well, hard to provision
- Customer/provider services:
  - DIFFSERV
  - still maturing
- Support for application: RTP and signalling