Motivation: Quality of Service

- Advantages of Packet-Based Transport (as opposed to circuit switched)
  - Flexibility
  - Optimal Use of Link Capacities, Multiplex-Gain for bursty traffic
- Drawbacks
  - Buffering/Queueing at routers can be necessary
  - Delay / Jitter / Packet Loss can occur
  - Overhead from Headers (20 Byte IPv4, 20 Byte TCP)

→ Protocol Improvements for Real-Time Applications necessary
  - Packet Prioritization (DiffServ, TypeOfService field in IPv4)
  - Resource Reservation (IntServ)
  - Traffic Engineering (QoS Routing, static/dynamic MPLS paths)
  - Connection Admission / Traffic Policing / Shaping
  - Header Compression
Real-time requirements: Parameters

- **User Plane QoS/Network Performance**
  - End-2-End Packet Delay (in particular interactive applications)
  - Delay Jitter
  - Packet Loss
  - Throughput/Goodput

- **Application Level QoS**
  - e.g. Video/Voice Quality (depending on codecs)

- **Signalling Plane**
  - Call Setup Delays
  - Fraction of blocked Calls

- **Reliability Aspects**
  - Failure probabilities of entities
  - Downtime distribution

- **Behavior at Handover**
  - Dropped Calls
  - Delayed / Lost packets

Focus here

See other lectures (Wireless Networks II and III)
Extended layered communication model

- Ultimate goal of RT service provisioning: user satisfaction
- Focus here: network aspects, i.e. L2-4

Relevant functionalities:
- PHY Layer
  - Bit/Symbol transmission → Throughput
  - Symbol error probabilities (channel conditions, interference)
  - Propagation delays

  L3: Network Layer: IP
  L2: MAC/LLC
  L4: Transport: TCP, UDP, RTP/UDP
  Application (L5)
  Session Control, e.g. SIP
  Middleware
  User Interface

  L1: PHYS

  Network QoS
  Application QoS
  User perceived QoS

Relevant Functionalities (cntd.)

- Link Layer (L2)
  - Medium Access Delays
  - Collisions/unsuccesful transmissions
  - Fragmentation
  - Forward error correction (FEC) and error detection (CRC)
  - Link-layer Retransmission Mechanisms (ARQ)
  - L2 scheduling, switching, buffering

- Network Layer (L3)
  - Path selection (routing)
  - Processing delays (e.g. for routing table lookup)
  - L3 buffering, scheduling, buffer management (RED)
  - [L3 fragmentation]
Relevant Functionalities (cntd.)

- Transport Layer (L4)
  - Multiplexing/de-multiplexing (UDP/TCP)
  - Error detection/checksums
  - In-order delivery, sequence numbers (TCP)
  - Acknowledgements and Retransmissions (TCP)
  - Flow/Congestion Control (TCP)

- Application Layer/Codecs
  - FEC/CRC
  - Application Layer Retransmissions
  - Application Layer sequence numbers

All Layers
- Increased volume due to headers

Internet Protocol (IP)

Internet Protocol IP, IPv4:
- Layer 3 Protocol (Network Layer): implemented in hosts and routers
- Packet (IP datagram) transmission between two hosts
  (variable packet size up to 65535 bytes, often restricted by Layer 2 protocols)
- Routing using 32 bit addresses (v4)
  - Normally based on destination address only!

- Real-time affecting properties
  - Packet duplications
  - Packet reordering
  - Packet loss
  - Fragmentation

- Real-time relevant functionalities
  - Scheduling in routers
  - Buffer management
  - Route selection

![IP datagram](image)
Example: Gaming Application

Types of Games
- Real Time Strategy
- Massively Multiplayer Online Role Playing Game
- First Person Shooter (FPS)
  - Considered here: Counter-Strike

Counter-Strike: QoS Requirements

- High requirements on
  - Delay (RTT<60ms)
  - Jitter
- Medium Requirements on
  - Packet Loss (<3%)
  - Bandwidth consumption rather low (22 players<1Mb/s)
Content

1. Motivation & Background
   • Real-Time applications
   • Parameters

2. Layering Models Revisited
   • Layers and their real-time relevant functionalities
   • Application Layer: Example Gaming

3. Real-Time Transport Protocol (RTP)
   • RTP Header, RTP Functionality
   • Control Protocol: RTCP

4. QoS Provisioning on the IP Layer
   • Overprovisioning
   • Differentiated Services
   • Integrated Services, RSVP
   • QoS Routing

5. Summary and Outlook

TCP and real-time applications

Discussion: TCP properties
• Reliable transmissions may cause additional delays (time-out + additional retransmission)
  – packet losses often less harmful for RT applications than retransmission delays
• Congestion control mechanisms control sending rate
  – not acceptable for RT applications
• No support of multi-casting

Problems of TCP over wireless
• Large RTTs cause long connection set-up times and slow transmission rates in slow-start
• Packet losses due to wireless link properties trigger TCP congestion control mechanisms
• RTT variations (e.g. Due to Layer 2 retransmissions) could cause TCP time-outs
**User Datagram Protocol UDP**

- User Datagram Protocol UDP (RFC 768)
  - Connectionless
  - Unreliable
  - No flow/congestion control
- Functionalities
  - (De-)Multiplexing: Port Numbers
  - Simple additive checksum → error detection
  - Length field

**Real-Time Transport Protocol, RTP (RFC3550)**

- Supports multi-party multimedia conferences
- Uses UDP and multicasting capabilities of IP
- Provides Time-stamps, sequence numbers
- Supports:
  - Codec description (in separate profile description, e.g. RFC3551)
  - Synchronization
  - Mixing & codec translation of multimedia streams
- Header

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Ver P X CC M PTYPE Sequence Number
| Timestamp | Synchronization Source Identifier | Contributing Source ID | ...
```
RTP details: Header Fields 1

- Version (2 bits)
  - currently 2
- P bit (1 bit)
  - 1 if zero padding follows the payload
- X bit (1 bit)
  - 1 if optional header extension is present
  - depends on application type
- CC: source count (4 bits)
  - maximum of 15 contributing sources is supported
- M: Marker bit (1 bit)
  - application dependent
  - mark events in data stream
  - e.g. frame start in video streams
- PTYPE: Payload Type (7 bits)
  - determines detailed interpretation of header and contents

RTP details: Header Fields 2

- Sequence Number (16 bits)
  - identifies RTP packets
  - initial sequence number chosen randomly for each session
- Timestamp (32 bits)
  - time at which the first octet of digitized data was sampled (relative)
  - random choice of initial time stamp
  - continuously incremented
  - clock granularity depends on application
- Synchronization Source Identifier (32 bits)
  - identifies the source of a stream
  - real source, mixer or translator
  - identification collisions resolved by protocol
- Contributing Source ID (variable size, CC x 32 bits)
  - list of source identifiers that contributed the samples mixed together by a mixer
RTP: Functional Entities

- **Mixer**
  - receives one or more RTP streams (contributing sources)
  - possibly changes data format (transcoding)
  - combines packets and forwards
  - Timing adjustments  \( \rightarrow \) mixer is new synchronizing source
- **Translator**
  - Forward RTP packets without changing synchronization
  - E.g. Encoding converters (without mixing), replicators from multicast to unicast
- **Monitor**
  - Receives RTCP packets, in particular receiver reports
  - Estimates QoS, fault-diagnosis, long-term statistics
  - Integrated in applications in session or separate application (3rd party monitor)

RTP: Timestamps

- **Wallclock time**
  - Absolute date and time
  - Using Network Time Protocol (NTP) format
    - 64 bit unsigned fixed-point number (32 bit integer part)
    - Representing seconds since 0h, Jan. 1, 1900
  - RTP uses only timestamp differences (no ambiguity as long as within 68 years)
- **RTP timestamps**
  - Measured in clock-ticks of certain resolution
  - Clock frequency depends on payload format (e.g. defined by sampling period)
  - Initial value should be random
- Linking of RTP timestamps to reference (wallclock) time in RTCP Sender Reports
- Synchronisation of sources (e.g. using NTP) recommended but not mandatory
Real-time transmission control protocol: RTCP

- Control Protocol for RTP flows
- Functionalities
  - Monitor QoS
  - Convey participant information
  - Rate adaptation for Receiver Reports (scalability, very few up to thousands of participants)
- Message Types
  - 200 Sender Report
  - 201 Receiver Report
  - 202 Source Description (including canonical name, CNAME)
  - 203 BYE
  - 204 Application specific

RTCP Sender Report, see Sect. 6.4-6.7 of RFC 3550 for other packet types

Content

1. Motivation & Background
   - Real-Time applications
   - Parameters
2. Layering Models Revisited
   - Layers and their real-time relevant functionalities
   - Application Layer: Example Gaming
3. Real-Time Transport Protocol (RTP)
   - RTP Header, RTP Functionality
   - Control Protocol: RTCP
4. QoS Provisioning on the IP Layer
   - Overprovisioning
   - Differentiated Services
   - Integrated Services, RSVP
   - QoS Routing
5. Summary and Outlook
**Over-Provisioning**

- Design network to be able to deal with worst-case traffic scenario
- **Advantage:**
  - no impact on architecture, protocols and user equipment
  - simplicity
- **Problems:**
  - Traffic depends on number of active users, user mobility, type of application, daily utilization profile → difficult forecasting
  - Data traffic tends to be very bursty (even ‘self-similar’) → waste of resources if planned for worst-case scenario → can be very expensive
  - Unforeseeable events can occur (new applications; changes in user behavior, e.g. always-on)

---

**Differentiated Services (DiffServ)**

- Basic Idea: reduce queueing delay/loss for critical traffic by preferential treatment at routers (multiple queues)
  → improve per-hop transmission behavior
- Packets marked by DiffServ Code Points (DSCPs, 6bit)
- Various scheduling disciplines at routers possible (e.g. static priority, weighted fair queueing)
- **Advantage:** Simple and scalable
- **Problem:** No performance guarantees unless used in conjunction with connection admission and traffic shaping/policing at ingress routers
**DiffServ Code Points (DSCP)**

<table>
<thead>
<tr>
<th>Differentiated Services (DS) Byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per Hop Behaviour</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version</th>
<th>IHL</th>
<th>TOS</th>
<th>Total Length</th>
<th>Identification</th>
<th>Flags</th>
<th>Fragment Offset</th>
<th>Source Address</th>
<th>Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version</th>
<th>Traffic Class</th>
<th>Flow Label</th>
<th>Payload Length</th>
<th>Next Header</th>
<th>Hop Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Source Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Destination Address</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**IPv4**

**IPv6**

**DiffServ: Influencing QoS**

- **At Border Routers**
  - Traffic classification and Marking $\rightarrow$ DiffServ Class (e.g. EF, AF, BE)
  - Traffic Policing/Shaping/Conditioning, e.g.
    - (Token) Leaky Bucket
    - Time-sliding window 3 color marker
      - Two thresholds: Committed Information Rate, Peak Information Rate
      - Time-sliding window for measurement of average rate
        $$ \Lambda_T = \Lambda_{T-1} + \Delta \sigma + \Delta t_i$$

- **At Interior DiffServ Router**
  - Scheduling: Strict Priority, Weighted Fair Queueing, etc.
  - Buffer Management, e.g. Random Early Drop RED, RIO
    - Possibly different drop precedence
**Integrated Services (IntServ) / RSVP**

- **Fundamental Idea:** Reserve necessary resources for each traffic flow along its transmission path, which requires:
  - Connection Admission Control (CAC): traffic specification + info about available resources at router → admission decision (if no, then re-routing)
  - Packet Classification: which flow does it belong to?
  - Packet Scheduling: make sure, flow obtains resources as specified

---

**IntServ: functionalities**

- **Connection/Call Admission Control (CAC)**
  - Easy for constant bit rate (CBR) flows
  - Difficult tasks for bursty traffic
    - Alternatives:
      - Peak-Rate Allocation → no multiplex gain
      - Mean-rate allocation → large delays and losses possible
      - Intermediate solution: effective bandwidths
        - Frequently based on limit theorems
          - Large deviations theory
          - High multiplex degrees
  - Packet Classifier
    - Flow specifications by so-called ‘filters’
    - Specifies ranges of value for L3/L4 header fields
  - Packet Scheduler
    - Multiple queues
    - Scheduling principles: WFQ, strict priority, EDF, ...
    - In addition: buffer management, e.g. RED (see later)
**IntServ: Signalling, RSVP**

- Signalling by Resource Reservation Protocol (RSVP)
  - Path Message: sender initiated, description of traffic parameters (Tspec) and path
  - Resv Message: receiver initiated, causes connection admission/reservation along path; specifies QoS parameters (Rspec)
  - Other messages for reservation teardown and error treatment
- Principles
  - In-path signalling
  - Multi-cast support
  - Soft-State concept: periodic refresh of reservation required
- **Advantages:**
  - Fine Granularity: per flow treatment, flexible set of QoS parameters
  - Able to provide QoS guarantees (if admission, classification, scheduling is performed correctly)
- **Disadvantages**
  - Scalability problem: management of state for each single flow
  - Complexity (already connection admission can be complex, e.g. effective bandwidths, etc.)

---

**IntServ: RSVP Messages I**

- **Path Message**
  - Tspec: Traffic specification
    - Token Bucket Rate
    - Token Bucket Size
    - Peak Data Rate
    - Minimum Policed Unit
    - Packet Size
  - Adspec: Network Resources on Path
    - Non QoS Hop-count
    - Available Path Bandwidth
    - Minimum Path Latency
    - Path MTU
  - Sender Template: Filter Specification
    - IP source address, protocol type, port number, etc.
  - Previous router on path
**IntServ: RSVP Messages II**

- **Resv Message**
  - Next hop in path (receiver → sender)
  - Flow-spec
    - Tspec
    - Rspec
  - List of Filter Specs (description of sender for which the reservation is intended)
  - Reservation style
    - Wildcard filter: shared, one reservation for all senders
    - Fixed filter: distinct, one per sender
    - Shared explicit: one reservation for specified list of senders

---

**RSVP Messages: FlowSpec (Controlled Load)**

<table>
<thead>
<tr>
<th>31</th>
<th>24 23</th>
<th>16 15</th>
<th>8 7</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 (a)</td>
<td>reserved</td>
<td>7 (b)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5 (c)</td>
<td>0</td>
<td>reserved</td>
<td>6 (d)</td>
</tr>
<tr>
<td>3</td>
<td>127 (e)</td>
<td>0 (f)</td>
<td>5 (g)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Token Bucket Rate [r] (32-bit IEEE floating point number)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Token Bucket Size [b] (32-bit IEEE floating point number)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Peak Data Rate [p] (32-bit IEEE floating point number)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Minimum Policed Unit [m] (32-bit integer)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Maximum Packet Size [M] (32-bit integer)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Reservation Types:**
  - Guaranteed Service: Bandwidth and Delay Guarantees, No Loss
  - Controlled Load: Only Bandwidth Guarantee

(a) - Message format version number (0)
(b) - Overall length (7 words not including header)
(c) - Service header, service number 5 (Controlled-Load)
(d) - Length of controlled-load data, 6 words not including per-service header
(e) - Parameter 127, parameter 127 (Token Bucket TSpec)
(f) - Parameter 127 flags (set)
(g) - Parameter 127 length, 5 words not including per-service header
### Additional Issues in QoS signalling

- Inter-domain signaling
- Off-path signaling
- Arbitrary placement of initiator and receiver
- Bi-directional signaling/ sender-initiated signaling
- Mobility support
- Implementation size & complexity (own transport protocol on top of IP, multicast support, etc.)
- How to secure RSVP in a real-world environment

### QoS signalling scenarios

**IN-Path**

- State kept at more than two entities.
- Protocol requires interaction with other protocols (routing, security, AAA, mobility, etc.)
Traffic Engineering (TE)

- TE = distribute traffic over network links in order to avoid congestion
- IP routing (OSPF, IS-IS, RIP, etc.)
  - Based on destination IP address
  - No possibility for distinguishing traffic classes
  - Link costs normally statically assigned (sometimes even hop-count used)
  - Modification of link costs possible, but implications on link utilizations not straightforward
- Alternatives
  - QoS Routing: Use QoS parameters for path selection
  - Establishment of explicit paths
    - Automatically
    - Via network management
    - Approaches:
      - Tunneling: e.g. L2TP, PPP
      - Multi-Protocol Label Switching (MPLS)

Traffic Engineering: Time-scales

- In traditional use: traffic engineering and configuration of link costs for routing done via network management → time-scales of minutes to hours (in best case)
- Shorter time-scales (flow-level, even packet level) via extensions, e.g. QoS routing
**QoS routing: Steps**

- Add QoS relevant information to link state advertisements (in addition to static link costs and connectivity relation)

- Two approaches
  - Source Routing: Compute full routes
  - Distributed Routing: determine next hop

- Modify path metrics using QoS parameters
- Constraint-based routing: eliminate certain paths not meeting constraints (e.g. Minimal bandwidth req.)

**BUT:** increased complexity, path selection in some cases np-complete!

---

**QoS routing: example**

- OSPF-TE: LSAs advertise
  - Cost
  - Residual bandwidth
  - Delay of links
- Widest-shortest path algorithm:
  - Among all paths with sufficient bandwidth
  - choose among those with the lowest hop count
  - If there are several feasible paths with identical hop count, choose the one with the highest residual bandwidth.
- Computation of routing tables at each node using a modified Bellman-Ford algorithm
Summary

1. Motivation & Background
   • Real-Time applications
   • Parameters
2. Layering Models Revisited
   • Layers and their real-time relevant functionalities
   • Application Layer: Example Gaming
3. Real-Time Transport Protocol (RTP)
   • RTP Header, RTP Functionality
   • Control Protocol: RTCP
4. QoS Provisioning on the IP Layer
   • Overprovisioning
   • Differentiated Services
   • Integrated Services, RSVP
   • QoS Routing
5. Summary and Outlook

Topics not treated here

• QoS aware link-layer protocols (802.1q, etc.)
• Traffic Classes (ATM, UMTS, etc.)
• Details of scheduling methods (WFQ, etc.)
• Details of buffer management (RED, etc.)
• (Token) Leaky Buckets
• Multi-Protocol Label Switching, MPLS (and signalling protocols, e.g. CR-LDP)
• QoS in wireless technologies
• End-to-end QoS signalling in UMTS (SIP/IMS)
• Coupling of Mobility support and QoS
• Service Level Agreements (SLAs)
• Performance Models

[and many more…]
Acknowledgements/References

- Student work
  - Raimund Brandt (TU Munich, Seminar)
- Presentations of H. Schulzrinne (Columbia University) and H. Tschofenig (Siemens Corporate Technologies)
- InfotechLecture notes: IP Based Networks and Applications, Chapter 3 (J. Charzinski), www.jcho.de/jc/IPNA
- Tutorial: IP Technology in 3rd Generation mobile networks, Siemens AG (J. Kross, L. Smith, H. Schwefel)