Quality of Service in the Internet

Problem today:

• IP is packet switched, therefore no guarantees on a transmission is given (throughput, transmission delay, …): the Internet transmits data Best Effort

• But: many applications need a certain Quality of Service (QoS) to achieve the results expected by the user:

<table>
<thead>
<tr>
<th>Application</th>
<th>Reliability</th>
<th>Delay</th>
<th>Jitter</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-Mail</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>File Transfer</td>
<td>high</td>
<td>low</td>
<td>low</td>
<td>medium</td>
</tr>
<tr>
<td>Web Access</td>
<td>high</td>
<td>medium</td>
<td>low</td>
<td>medium</td>
</tr>
<tr>
<td>Remote Login</td>
<td>high</td>
<td>medium</td>
<td>medium</td>
<td>low</td>
</tr>
<tr>
<td>Audio on Demand</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>medium</td>
</tr>
<tr>
<td>Video on Demand</td>
<td>low</td>
<td>low</td>
<td>high</td>
<td>high</td>
</tr>
<tr>
<td>IP Telephony</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>low</td>
</tr>
<tr>
<td>Video Conference</td>
<td>low</td>
<td>high</td>
<td>high</td>
<td>high</td>
</tr>
</tbody>
</table>

QoS Parameters

Throughput (bytes/s)

- Which minimum/maximum/average data rate is necessary?

Transmission delay (ms)

- Which maximum delay is tolerable?

Jitter (ms)

- Which fluctuations in the transmission delay are tolerable?

Availability (%)

- With which probability the communication service is available?

Keeping the QoS

Mechanism: regard a flow of packets as stream, try to enforce quality requirements for the stream as a whole.

- Traffic Shaping, e.g. Leaky bucket and token bucket
- Over-provisioning: increased router capacities, buffers and line capacity
- Buffering: temporary store the data within the receiver. Thus the delay is increased, but the jitter is lowered (audio/video on demand)

Traffic Shaping: Leaky Bucket Algorithm

Problem: Peak traffic is not considered. Data can be lost, even if sufficient capacity is available in the network.
Traffic Shaping: Token Bucket Algorithm

Before

Host

Network

Per time unit a token flows into the bucket

Bucket collects tokens

After

Host

Network

Per token a packet is sent

Inside Routers: Packet Scheduling

Determine the order, in which routers pass on arriving packets.

- **FIFO (First In, First Out)**
  The packets will be transferred regarding their arrival; if the buffer is full, further packets are rejected. This is usual practice in routers.

- **Priority Queuing**
  Basing on a characteristic (bit in the header, destination address,…) a priority is assigned to a packet. For each priority an own buffer exists. The highest priority, non-empty queue is always processed first.

Packet Scheduling

- **Round Robin**
  Scan all queues in turn and in each case send a packet from each non empty queue

- **Weighted Fair Queuing**
  Combine Round Robin with prioritizations of the individual queues

Weighted Fair Queuing

Traffic Shaping in a Router

A router has several queues for an outgoing links. The queues are scanned cyclically for determining which one is allowed to send. The scanning is made byte by byte of the queues; if a packet is completely scanned, it is sent. Weighted means that certain queues get larger portions of the scanning time than others.

Router with five packets for line O

<table>
<thead>
<tr>
<th>Packet</th>
<th>Sending time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>16</td>
</tr>
<tr>
<td>B</td>
<td>11</td>
</tr>
<tr>
<td>C</td>
<td>13</td>
</tr>
<tr>
<td>D</td>
<td>17</td>
</tr>
<tr>
<td>E</td>
<td>18</td>
</tr>
</tbody>
</table>

Sending time of the packets
Guaranteeing Delay: Resource Reservation

- Problem with all these techniques: packets can take different ways
- Necessary: establish a path through the network and reserve fixed amounts of network capacity, buffer capacity and CPU time with the routers.
- Standardized by the IETF as Resource Reservation Protocol, RSVP

RSVP is not a routing protocol, but only an “addition” to such one. A description of the requirements of the receiver in form of a flow specification (Level of the QoS) is needed. Categories of flow specification are

1.) Best Effort: Take all free resources you can get
2.) Rate Sensitive: a guaranteed (minimal) data transmission rate is necessary
3.) Delay Sensitive: a guaranteed (maximal) delay is necessary

RSVP can be used for Unicast and for Multicast

RSVP Procedure:

- The sender sends a RSVP Path Message to the receiver. Information about the routers on the network path are collected and communicated to the receiver. (Additionally, also flow specifications for the receiver are provided.)
- The receiver sends RSVP Reservation Messages along the collected path, which contain its flow specification. Every router on the path reserves resources accordingly.
- If a reservation is not possible with a router, an error message is sent back.
- As soon as the RSVP Reservation Message reaches the sender, the complete path is reserved, the sender begins to send its data.
- To delete outdated reservations, timeouts are defined. If the receiver does not refresh its reservation before expiration of the timer, it is deleted.

RSVP with Multicast:

(a) A path from receiver 3 to sender 1
(b) A second path for sender 2, because both data streams are independent
(c) For multicast, already existing reservations can be used

QoS in the Internet: Integrated Services

- 1995-1997 standardized by the IETF as an architecture for the transmission of multimedia streams
- Integrated Services (IntServ) uses RSVP for signaling of flow specifications as well as reservations and provides QoS for each data flow individually

- Thus the principle adds a connection-orientation to IP
Integrated Services

- Classes of QoS:
  - **Guaranteed**: Data rate, delay and reliability are guaranteed. Deviations do not occur, packets are not discarded. Suitable for intolerant real-time applications.
  - **Controlled Load**: "weak" guarantees, small deviations are possible. Principle: for data streams of this class the network behaves as best effort for an unloaded network. No guarantees are given, suitable for tolerant real-time applications.
  - **Best Effort**: as good as possible, normal Internet traffic.

- Problems:
  - **Scalability**: for each data flow a router has to maintain own flow specifications
  - How can **authorization** and **priority** be treated for a reservation request?
  - The QoS classes are not sufficient to differentiate reasonably between different types of data streams.

- Possible: use IntServ only “at the edge” of large networks, where only few data flows are present

QoS in the Internet: Differentiated Services

- Class-based approach (**Differentiated Services**, DiffServ): do without guarantees, only manage aggregated data streams

- Thus, the complexity in routers is shifted “at the edge” of the network, internal network routers can be kept simpler.

- Divide the network into domains. A domain is a part of the entire network, which supports DiffServ. It consists of access routers for the domain (Ingress routers, yellow) and internal routers (core routers, blue).

- The domain defines service classes, each data flow is assigned to such a class. With coming into the domain (at a **Ingress Router**), each packet is assigned a class, and in the network the forwarding bases on the class parameters (per-hop behavior). Looking only at the aggregated data flows, a better scalability as for IntServ is achieved.

Application of DiffServ with IP

- Use of the Type of Service field in IPv4 for the classification (**DSCP – Differentiated Service Code Point**). The DSCP value defines the per-hop behavior of the packet from one router to the next one.

  - **Version**: 4
  - **IHL**: length of the IPv4 header in 32-bit words
  - **Type of Service**: DSCP
  - **Total Length**: total length of the IP packet
  - **Identification**: unique identifier of the datagram
  - **Fragment Offset**: offset of the fragment within the datagram
  - **Protocol**: upper-layer protocol
  - **Header Checksum**: checksum of the IPv4 header and the upper-layer protocol header
  - **Source Address**: address of the source host
  - **Destination Address**: address of the destination host

  The code point defines the transmission class, which informs a router, how it has to treat the packet in forwarding.
### Service Classes

| DSCP | free |

For DiffServ, amongst others the following classes (for specifying transmission behavior) are defined at the moment:

- **Default Forwarding** realizes the usual Best Effort transmission
- **Expedited Forwarding** tries to emulate a rented line
- **Assured Forwarding** uses priorities and discarding probabilities used when an overload of a router is given

### Expedited Forwarding

Idea with Expedited Forwarding:
- there are “regular” and “expedited” packets. Expedited packets are passed on in such a way, as if there would be no or only little further traffic. A minimum data transmission rate is guaranteed.
- Routers can manage two separated queues for these packet types (Weighted Fair Queuing).
- Possible: low loss rate/delay/jitter as well as a guaranteed data transmission rate

### Assured Forwarding

- Improves differentiation: Definition of four priority classes with own resources (at the moment; there also is room for more classes)
- For each class, three probabilities for discarding a packet are defined: low, medium, high
- Thus: altogether 12 different service classes
- Principle:
  - The priority class determines the portion of the transmission capacity of the routers
  - During high load packets of lower priority would be discarded completely
  - Fairness: packets of each priority class should have chances to survive
  - Therefore definition of the probabilities for each class: by suitable selection of the probabilities, a small part of the lowest priority level still is still forwarded, while packets of higher priority classes are already discarded.
**Assured Forwarding**

1. Classification of the packets in service classes
2. Appropriate choice of DSCP tags
3. Bring the data streams in a form according to their flow specification. Exceeding the specifications leads to discarding data. If thereafter still too much data are present, discarding of packets in accordance with their probability
4. Weighted Fair Queuing in accordance to priority classes

**Label Switching: MPLS**

While the IETF worked on IntServ and DiffServ, also routing manufacturers continued to think ahead: assign each packet with a label, on which basis the routing decisions are made (instead of the destination address, like for ATM).

- Result: MultiProtocol Label Switching, MPLS
- The label is not placed in the IP header it is sent before it:

**MPLS**

Principle:
- The first packet of a data stream determines the path. The routers stores the label information of the packet and sends all further packets with this label over the same link (logical channel). Thus a connection-orientation is achieved.
- Routers decide on the basis the label, on which outgoing link a packet is forwarded. At the same time, also a new label is set, which “instructs” the next router.
- A router again can manage different queues, the belonging of a packet is determined by the QoS field (service class as in DiffServ).
- At the end of the network the label can be removed and the IP address is used for path choice.
- Difference to connection-oriented networks: there is no connection establishment.
- In hierarchical networks a packet can have several stacked labels for the hierarchies. In this case the S-bit is set (S = stackable)

**Routing in the Edge, Switching in the Core**

1. label request (IP)
2. label mapping (IP)
3. sending data

- The LANs need not to be modified, only enhancement of the routers inside the core network
- Path establishment using usual IP, similar to ATM
- At the edge networks: usual IP-based routing
- In the core network: enhanced Routers establish a path and use switching tables, not routing tables

Note: none of the approaches for QoS is commonly used in the Internet so far – momentarily everything is still in the experimental stage
### Why MPLS?

- **IntServ** cannot manage a large number of connections — only useful at the network edge with a low number of connections.
- **DiffServ** is better for the core network, but only aggregates all traffic to a few number of classes.
- ATM is purely circuit switched and does not perform very well for data transmission.
- IP is purely packet forwarding and not useful for multimedia data.

Thus: MPLS together with IP are kind of “hybrid” approach between packet forwarding and circuit switching: combine best of both worlds.

---

### Conclusion

TCP/IP reference model defines only one protocol for layer 3: IP.

- Connectionless transmission, data packets are forwarded hop-by-hop.
- Supported by routing protocols to determine the best way to a destination.
- ICMP for exchange of control messages.
- ARP for mapping of IP addresses to MAC addresses.

But: several problems with the “current” IPv4: address space, security, mobility, quality of service…

- Large number of additional protocols to deal with these problems: network address translation, IPsec, Mobile IP, IntServ, DiffServ, MPLS, …

Successor IPv6 could deal with the problems, but will it ever come?