Today:

- ATM
- MPLS
- QoS
- Smoothing
- Summary

**ATM: Asynchronous Transfer Mode**

**Goal:** Combine telecommunication and data networks in a single standard

**Facts:**
- The Internet carries data efficiently, but carries real-time traffic poorly.
- The telephone network carries voice clearly, but data inefficiently.

**Solution:** ATM. First proposed in ~1985 for "Broadband ISDN." ATM mostly failed as far as data networks are concerned.

**Key Attributes of ATM**

**Virtual Circuit switching:**
A stream first allocates a route called “virtual circuit.”
- Each switch on the route knows the VCI and its next hop, allocates buffers, bandwidth etc.

**Fixed-length small cells:**
The stream consists of a sequence of 53-bytes cells. (!)
- simplify switch design,
- allow fine-grain control of delay through a switch.

**Virtual Circuits**

Virtual Circuits are established hop-by-hop from end-to-end.
Virtual Circuit Identifiers

VC Table at Port 3:

<table>
<thead>
<tr>
<th>VC in</th>
<th>VC out</th>
<th>Port out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

VCI: Virtual Circuit Identifier (includes VPI)
HEC: header error control
PT: payload type (3 bits). Data/control? Congestion experienced? If user: last segment?
CLP: cell loss priority (1 bit)

Fixed-length cells

PAYLOAD DATA
HEC
VCI

48 bytes
5 bytes

48 bytes

VCI: Virtual Circuit Identifier (includes VPI)
HEC: header error control
PT: payload type (3 bits). Data/control? Congestion experienced? If user: last segment?
CLP: cell loss priority (1 bit)

AAL 5: Simple Data Service

ATM Adaptation Layers (AALs) provide a variable-length packet interface to ATM.

CRC Length PAD DATA

Segmentation

Reassembly

ATM Service Classes

- Constant Bit rate (CBR)
  - Uses leaky-bucket traffic descriptor.
  - Good for telephony, or emulation of circuit switching in general.
- Variable Bit rate (VBR)
  - Dual leaky-bucket regulated source.
  - Network guarantees end-to-end delay (for VBR-rt)
- Available Bit rate (ABR)
  - Includes Congestion-control mechanisms similar to TCP.
- Unspecified Bit rate (UBR)
  - Best-effort: whatever is left-over.
The adoption of ATM

- ATM was designed to replace the Internet by offering end-to-end QoS, high-performance switches, and integration of the voice and data networks.
- It didn’t happen as planned.
  » Complex standards.
  » Momentum and simplicity of IP and the Internet.
  » Signaling to create Virtual Circuits complex and not robust.

MPLS: an idea from ATM

- MPLS: Multi-Protocol Label Switching
- A forwarding scheme designed to speed up IP packet forwarding (RFC 3031)
- Idea: use a fixed length label in the packet header to decide packet forwarding
  » Label carried in an MPLS header between the link layer header and network layer header
- Support any network layer protocol and link layer protocol

Basic MPLS terminology

- LSR: An MPLS capable router (label switching router)
- Forwarding Equivalence Class (FEC): A subset of packets that are treated the same way by an LSR
- A packet is assigned to an FEC at the ingress of an MPLS domain

MPLS Operation

- Ingress LSR of an MPLS domain: insert an MPLS header before the packet is forwarded
  » Label in the MPLS header encodes the packet’s FEC
- At subsequent LSRs
  » The label is used as an index into a forwarding table that specifies the next hop and a new label.
  » The old label is replaced with the new label, and the packet is forwarded to the next hop.
- Egress LSR strips the label and forwards the packet to final destination based on the IP packet header
Label Switched Path (LSP)

- For each FEC there is a specific LSP
- To set up an LSP, each LSR assigns an incoming label to the LSP for the corresponding FEC
- A forwarding table is constructed as the result of label distribution.
  - A backup LSP is set up in advance from the source LSR to the destination LSR of the primary LSP.
    - The backup LSP is link and node disjoint with the primary LSP

Label Stacking

- A packet may carry multiple labels, organized as a last-in-first-out stack
- A label may be added to/removed from the stack at any LSR
- Processing always done on the top label
- Label stacking allows different primary LSPs to use the same bypass tunnel for failure protection
  - Bypass tunnel: a LSP used to protect a set of LSPs passing over a common facility.

Local Protection Using Label Stacking

When a failure occurs:
- LSR at the beginning of the tunnel
  - Replaces old label as usual (for last tunnel node)
  - Pushes the bypass tunnel’s label onto the label-stack of the redirected packets.
  - Switches packets to the bypass tunnel
- LSR at the end of the tunnel will
  - Pop the bypass tunnel’s label
  - Examine the top label to determine the protected LSP that the packet is to follow.

Quality of Service (QoS)
QoS: Quality of Service

- Current Internet: **Best Effort** promise
  - No hard guarantees
  - Insufficient for “real” applications:
    - mission critical
    - real-time
- Future Internet (and current telecom):
  **Guaranteed** performance
  - Several Quality of Service parameters

Typical QoS Parameters

ATM (and IPv6) standards support:
- packet loss rate
- bandwidth
- end-to-end delay
- delay jitter

QoS setup: A Contract

- Incoming connection specifies requirement
- If network can meet requirement:
  - User commits to maximal usage
  - Network commits to minimal allocation
- Otherwise, connection is rejected
- During connection, network and user may monitor each other (and sue if contract violated...)

Some applications would like bounded delay

- Multimedia Applications
  - One-to-many streaming stored audio or video.
  - Interactive streaming audio or video.
- Other delay-sensitive applications
  - Real-time control.
  - Games
- Other delay-sensitive applications
  - Premium Internet/web access
Why care?
Case in Point: Movies

- Huge bandwidth consumer
- Without compression: scary
  - 500×300 pixels, byte per pixel, 25 frames/sec: 30Mbps
- With compression: highly bursty
  - MPEG: temporal and spatial information. Frames vary in size, scenes vary in bandwidth rate
  - JPEG: spatial compression; MPEG: I, P and B frames.

What to do? (1)

- **Conservative rich**: reserve peak required bandwidth.
  - Wasteful, costly: ratio of peak to average rate may be over 10:1
- **Conservative poor**: compress to specified bandwidth

MPEG with constant bandwidth

- Practical solution: trade bandwidth for memory
  - With large storage space: trickle the stream, playback from memory
    - need huge space
    - how about live broadcast?

Why is it always blurry exactly when it starts getting interesting?
What to do? (3)

**Smoothing:** reserve more than average bandwidth; start playback a short time after transmission starts.

The Setting

Example: stored video

Assumptions:
- Server stores all N video frames
- Client with a playback buffer
- Constant delay, lossless channel

Goal: minimize peak bandwidth, #bandwidth changes

Lossless smoothing

$L_k$ - Data played out by the client by time $k$
$U_k$ - Data available at the server by time $k$
$U_k = L_{k+D}$

# bytes (cumulative) vs. time (in frames)
Lossless smoothing

$S$ - Transmission Schedule
- non decreasing
- $S_0 = U_0$, $S_n = L_n$
- $U \geq S \geq L$

![Graph showing cumulative bytes vs. time for server and client buffers.](image)

Rubber-band algorithm:
Extend a straight line as far as possible;
If hit $L$, set new start point on $U$ and vice-versa.
Gives the most "evenly distributed" bandwidth vector.

What's "evenly distributed?"

- Consider $X=(x_1, x_2, ..., x_n)$ and $Y=(y_1, y_2, ..., y_n)$ with $\sum x_i = \sum y_i$.
- Sort: $x_1 \geq x_2 \geq ... \geq x_n$ and $y_1 \geq y_2 \geq ... \geq y_n$.
- $X$ is majorized by $Y$ (is smoother than $Y$) iff for all $k \leq n$:
  \[ \sum_{i=1}^{k} x_i \leq \sum_{i=1}^{k} y_i \]
- Example: $X,Y= (4, 3, 3, 5, 1) (2, 2, 3, 1, 8)$.
  Sorted:
  $\text{Prefix sums:}$
  \[
  \begin{align*}
  X & \quad (4, 3, 3, 5, 1) & \quad (2, 2, 3, 1, 8) \\
  \text{Sorted:} & \quad (5, 4, 3, 3, 1) & \quad (8, 3, 2, 2, 1) \\
  \text{Prefix sums:} & \quad (5, 9, 12, 15, 16) & \quad (8, 11, 13, 15, 16)
  \end{align*}
  \]
  Hence $X$ is majorized by $Y$. Is $X$ majorized by $Y'= (2,2,2,2,8)$?
- If $X$ is majorized by $Y$, then $f(X) \leq f(Y)$ for any convex $f$ (e.g., max, variance...)

Experimental result

(a) Bandwidth plan with 1-megabyte buffer

The rubber-band algorithm produces a schedule which is minimal w.r.t. majorization.
But in real life: Delay Varies...

Lossy Smoothing

Fix buffer space, bandwidth, playback delay
- Source generates stream
- Server fills some of its buffer
  » may lose data by overflow
- Server starts transmitting
- Client fills some of its buffer
- Client starts playing out
  » may lose data also by underflow

Lossy Smoothing: Questions

- How much does the server hold before transmitting?
- How much does the client hold before playing out?
- What’s the best buffer space at the server? the client?
- What’s the best bandwidth to use?

Lossy Smoothing: Algorithm

- Server:
  » timestamp each frame with relative time.
  » Push out data at maximal possible rate without any delay.
- Client:
  » Delay first bit for $D$ time units
  » Then start playback according to timestamps
- Parameters:
  » server buffer size = client buffer size
  » buffer size = bandwidth x delay
Why?

Lemma: Suppose bandwidth is $C$ and buffer size is $B$. Then delay $D=B/C$ ensures minimum loss of data.

Proof: Clearly, server has minimal overflow. Client never overflows: in $D$ time units, can receive at most $D \cdot C=B$ bits. Also, client never underflows: each bit transmitted is delayed at most $B/C=D$ time units.

- For cases of no loss, get best resource utilization

What does this mean?

In cases where two parameters are given, can optimally choose third:
- For live broadcasts, delay and bandwidth may be fixed etc.
- In other cases, only one parameter is fixed (say bandwidth), and we can find most appropriate tradeoff

What to do? (summary)

Assume parameters are known.
- Server starts transmitting immediately
- Client waits for $B/C$ time units, then starts playing out
- Can negotiate parameters during setup

Looking back...

- IP:
  - addresses and tricks: DNS, ARP
  - RIP, OSPF, BGP
- Reliable transmission: ARQ
  - Sliding windows
  - Stop & Wait, Go back N, selective repeat
- TCP
  - Setup and teardown
  - RTT, ACK rules
  - Congestion control & avoidance
- Security: Secret and public key
  - RSA, authentication, KDC, SSL
- ATM, MPLS, SMTP, HTTP