Multimedia Communications

Multimedia Technologies & Applications

Prof. Dr. Abdulmotaleb El Saddik
Multimedia Communications Research Laboratory
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada

elsaddik @ site.uottawa.ca
abed @ mcrlab.uottawa.ca

Outline

- History of Internet
- Multimedia and the Internet Protocols
- Networking Protocols
  - TCP
  - UDP
  - IP v.4, IP v6
- Delivery Paradigms
  - Unicast
  - Multicast
  - Broadcast
- Hypertext, hypermedia
- Standard Generalized Markup Language (SGML)
- Transport Protocol Solutions
  - RTP
  - RTSP

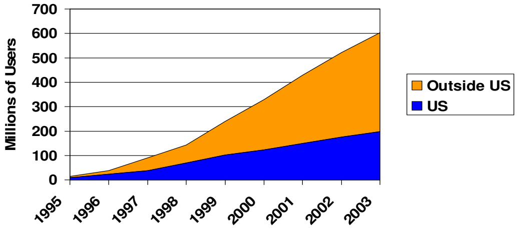
History of Internet

- It started as a research project to experiment with connecting computers together with packet switched networks.
- It was developed with funding and leadership of the Defense Department’s Advanced Research Projects Agency (ARPA).
- Development of TCP/IP
  - 1974: First full draft of TCP produced
  - 1978: TCP split into TCP and IP
- 1990: First ISP world.std.com

Who invented the Internet?

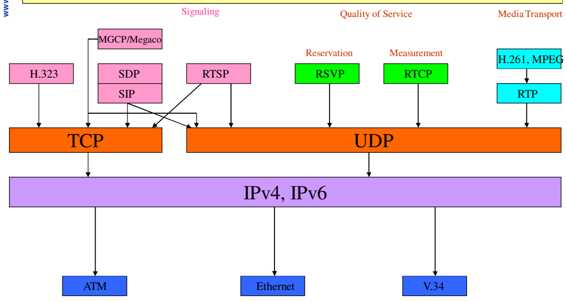
- ➢ **Unknown?**
  - ❖ No ☺
- ➢ **Leonard Kleinrock**
  - ❖ who did early work in packet switching?
- ➢ **Vint Cerf and Robert Kahn**
  - ❖ who defined the "Internet Protocol" (IP) and participated in the development of TCP?
- ➢ **Tim Berners-Lee**
  - ❖ Who developed HTTP to support a global hyper-text system he called the World Wide Web? (Internet vs the World Wide Web?)

People on the Internet (Worldwide)



Source: IDC

Multimedia & The Internet



Multimedia and the Internet

User support layers	7	Application	FTP, Telnet, SMTP, X-Windows, HTTP, …
Transport layer	4	Transport	TCP, UDP, TP4
Network support layers	3	Network	IP, ICMP, CLNP
	2	Link	Ethernet, X.25, FDDI, Token Ring PPP, SLIP
	1	Physical	RS232, V.35, 10BaseT

Client-Server Paradigm

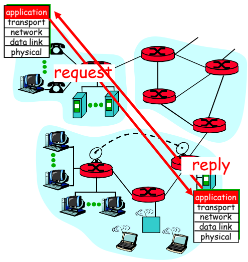
Typical network app has two pieces: *client* and *server*

Client:

- **initiates contact with server ("speaks first")**
- **typically requests service from server,**
- **for Web, client is implemented in browser; for e-mail, in mail reader**

Server:

- **provides requested service to client**
- **e.g., Web server sends requested Web page, mail server delivers e-mail**



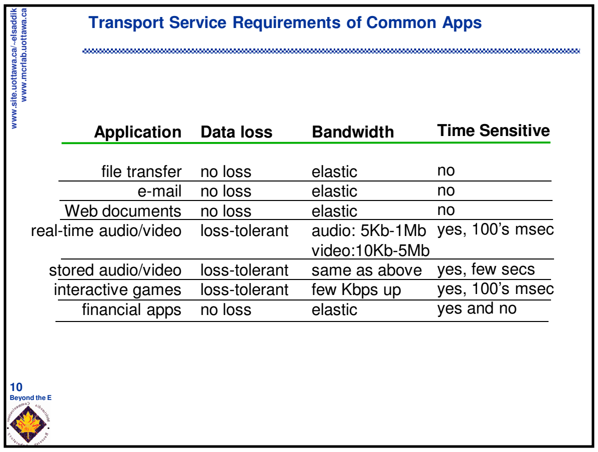
Application-Layer Protocols: How to Use the Transport Service?

API: application programming interface

- **defines interface between application and transport layer**
- **socket: Internet API**
  - **two processes communicate by sending data into socket, reading data out of socket**

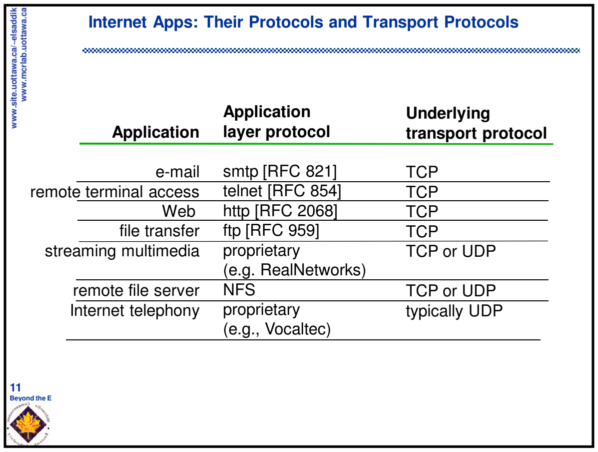
Q: how does a process "identify" the other process with which it wants to communicate?

- **IP address of host running other process**
- **"port number" - allows receiving host to determine to which local process the message should be delivered**



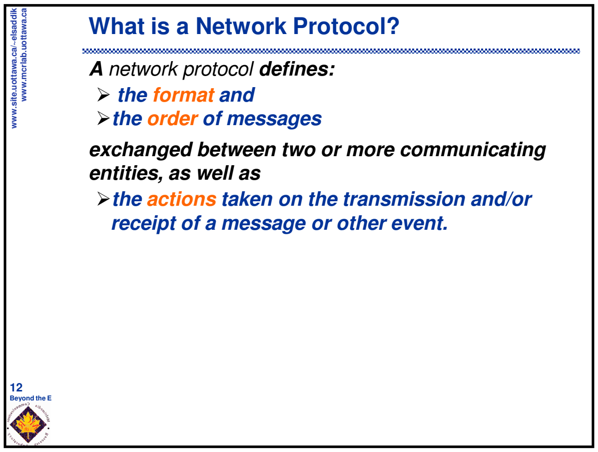
Transport Service Requirements of Common Apps

Application	Data loss	Bandwidth	Time Sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	no loss	elastic	no
real-time audio/video	loss-tolerant	audio: 5Kb-1Mb video:10Kb-5Mb	yes, 100's msec
stored audio/video	loss-tolerant	same as above	yes, few secs
interactive games	loss-tolerant	few Kbps up	yes, 100's msec
financial apps	no loss	elastic	yes and no



Internet Apps: Their Protocols and Transport Protocols

Application	Application layer protocol	Underlying transport protocol
e-mail	smtp [RFC 821]	TCP
remote terminal access	telnet [RFC 854]	TCP
Web	http [RFC 2068]	TCP
file transfer	ftp [RFC 959]	TCP
streaming multimedia	proprietary (e.g. RealNetworks)	TCP or UDP
remote file server	NFS	TCP or UDP
Internet telephony	proprietary (e.g., Vocaltec)	typically UDP



What is a Network Protocol?

A** network protocol **defines:

- ***the format and***
- ***the order of messages***

exchanged between two or more communicating entities, as well as

- ***the actions taken on the transmission and/or receipt of a message or other event.***

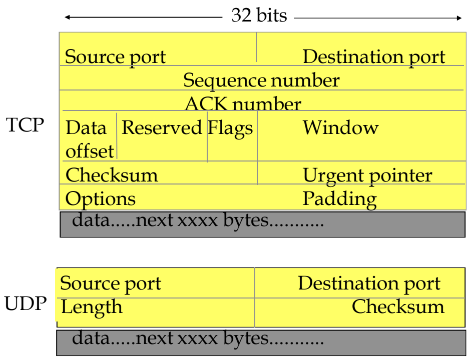
TCP

- Transmission Control Protocol
- TCP is stream oriented, designed to provide reliable, full-duplex communication between pairs of processes across a variety of reliable and unreliable networks and internets
- Multimedia communications do not always require full-duplex connections for continuous media transfer (e.g., TV broadcast)
- To achieve reliable, in sequence delivery of stream of bytes over an unreliable underlying datagram service, TCP makes use of retransmissions on timeouts and positive ACKs
- For multimedia, positive ACKs cause substantial overhead
- Sequence numbering is used, because of retransmissions
- TCP flow control makes use of a “window” technique
- Functionally equivalent to Class 4 ISO Transport protocol
- TCP not suitable for video and audio, because retransmissions may violate deadlines in continuous media streams
- TCP was designed for non-real-time reliable applications (e.g., file transfers) where it performs the best

User Datagram Protocol (UDP)

- UDP above IP can be used as a simple, unreliable connection for medium transport
- offers only multiplexing and checksumming, nothing else
- higher-level protocols using UDP must provide their own retransmission, packetization, reassembly, flow control, congestion control, etc.
- many multimedia applications use UDP because of some real-time transport property, although protocol data units may get lost
- in general, UDP is not suitable for continuous media streams, since it does not provide notion of connections and thus no QoS guarantees

TCP, UDP Header Formats



TCP

Source port			Destination port
Sequence number			
ACK number			
Data offset	Reserved	Flags	Window
Checksum			Urgent pointer
Options			Padding
data.....next xxxx bytes...........			

UDP

Source port	Destination port
Length	Checksum
data.....next xxxx bytes...........	

TCP vs. UDP

TCP service:

- *connection-oriented:* setup required between client, server
- *reliable transport* between sending and receiving process
- *flow control:* sender won't overwhelm receiver
- *congestion control:* throttle sender when network overloaded
- *does not provide:* timing, minimum bandwidth guarantees

UDP service:

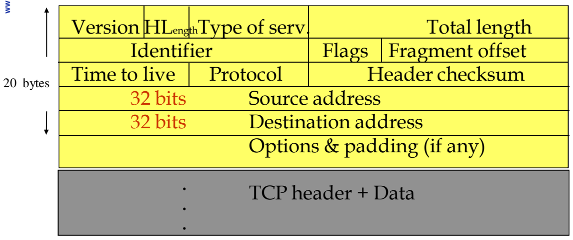
- *unreliable data transfer* between sending and receiving process
- *does not provide:* connection setup, reliability, flow control, congestion control, timing, or bandwidth guarantee

---

Internet Protocol (IP)

- TCP tells IP the Internet address of the destination host and that is all that IP is concerned about.
- IP's job is to find a route for the datagram and get it to the destination
- IP adds its own header , to allow gateways (routers) to forward the datagram.

IP protocol data unit



32 bits

Version | HLength | Type of serv. | Total length

Identifier | Flags | Fragment offset

Time to live | Protocol | Header checksum

32 bits Source address

32 bits Destination address

Options & padding (if any)

TCP header + Data

20 bytes

Addressing

Hierarchical addressing scheme

- Why?

Conceptually …

- IPaddress =(NetworkAddress,HostAddress)

IP establishes global address space,

- allowing every network in the Internet to be identified
- 5 address classes

Address Classes

Class A 0 Net(7b) Host (24 bits) (Very big networks)

Class B 10 Net (14 bits) Host (16 bits) (Normal large organiz.)

Class C 110 Net (21 bits) Host (8 bits) (Small networks)

Class D 1110 Multicast Address

Class E 1111 Reserved

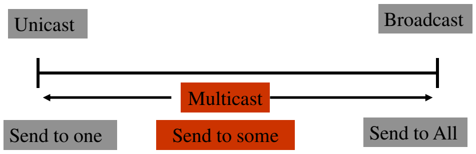
IP addressing details

- Internet addresses are 32-bit numbers normally written as 4 octets (decimal), e.g. 137.122.20.22
- Net address may be devided in subnet address:
  - Uof Ottawa Net address: 35194 (137.122) [class B address]
  - Engineering Subnet: 20
  - Host: 22

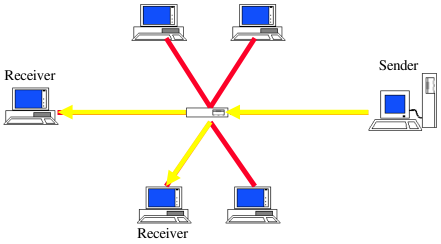
Class C Allocation Rules

Start	End	Location
194.0.0.0	195.255.255.255	Europe
198.0.0.0	199.255.255.255	North America
200.0.0.0	201.255.255.255	Central and South America
202.0.0.0	203.255.255.255	Asia and the Pacific

A Spectrum of Paradigms



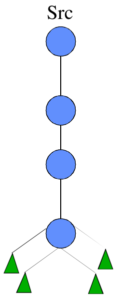
Unicast



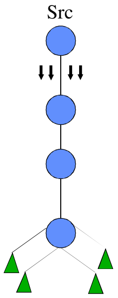
- Application sends one copy of each packet to each client’s unicast address
- Significant scaling restrictions, especially if the group is large
- The same information has to be carried multiple times, even on shared links
- Total B/W required for a stream = Stream (bps) x # of clients

---

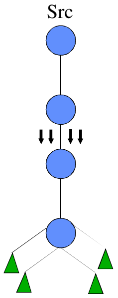
Unicast



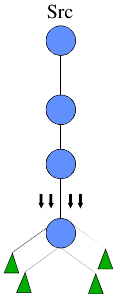
Unicast



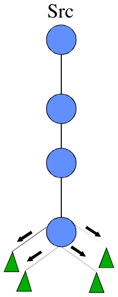
Unicast



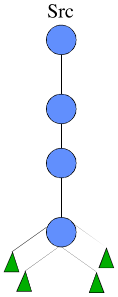
Unicast



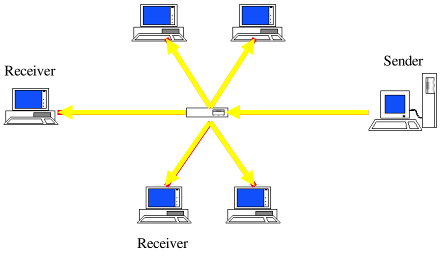
Unicast



Unicast

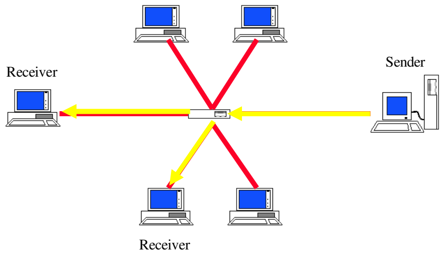


Broadcast



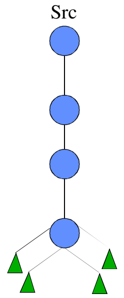
- Application sends one copy of each packet and addresses it to a broadcast address that all devices listen to.
- Simple for the applications to implement
- Not very efficient. All devices on a segment receive the packets.
- Layer 3 device (router) must either stop broadcasts or transmit broadcasts everywhere.
- Transmitting the broadcasts everywhere can be inefficient if only a small group actually needs to see the packet

Multicast

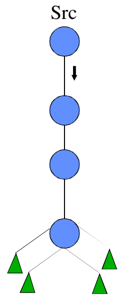


- Application sends one copy of each packet and addresses it to a group address
- The client decides whether or not to listen to the multicast (Group) address.
- Multicasting is helpful in controlling network traffic and curbs network and host processing by eliminating traffic redundancy.
- Multicast SCALES !!

Multicast

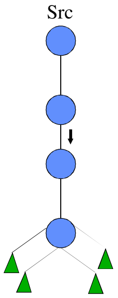


Multicast



Multicast

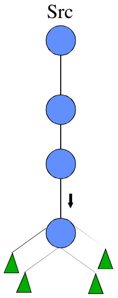
www.site.uottawa.ca/~elsaddik
www.mcrlab.uottawa.ca



Beyond the E

Multicast

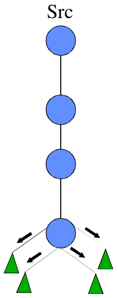
www.site.uottawa.ca/~elsaddik
www.mcrlab.uottawa.ca



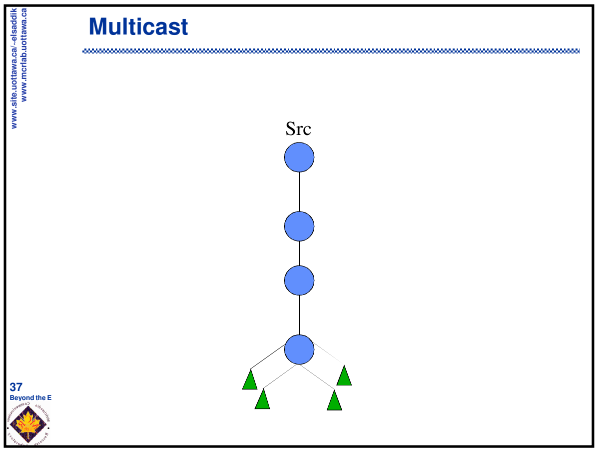
Beyond the E

Multicast

www.site.uottawa.ca/~elsaddik
www.mcrlab.uottawa.ca



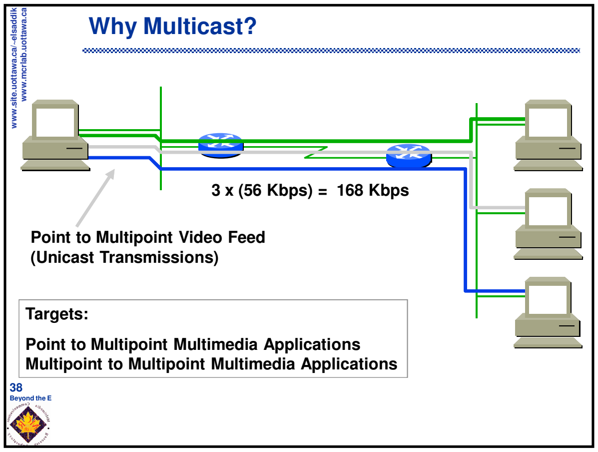
Beyond the E



Multicast

www.site.uottawa.ca/~elsaddik
www.mcrlab.uottawa.ca

37
Beyond the E



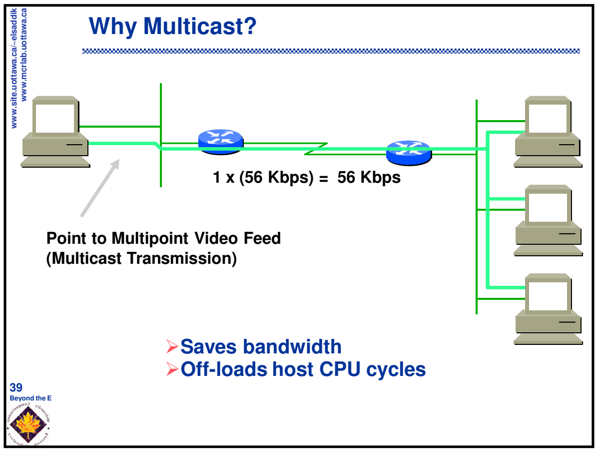
Why Multicast?

Point to Multipoint Video Feed
(Unicast Transmissions)

Targets:
Point to Multipoint Multimedia Applications
Multipoint to Multipoint Multimedia Applications

www.site.uottawa.ca/~elsaddik
www.mcrlab.uottawa.ca

38
Beyond the E



Why Multicast?

Point to Multipoint Video Feed
(Multicast Transmission)

- Saves bandwidth
- Off-loads host CPU cycles

www.site.uottawa.ca/~elsaddik
www.mcrlab.uottawa.ca

39
Beyond the E

Multicast Applications

- News/sports/stock/weather updates
- Distance learning
- Configuration, routing updates, service location
- Pointcast-type “push” apps
- Teleconferencing (audio, video, shared whiteboard, text editor)
- Distributed interactive gaming or simulations
- Email distribution lists
- Content distribution; Software distribution
- Web-cache updates
- Database replication

Next Generation IP : IPv6

- IPv4 (current IP) has 32-bit addresses
- IPv6 will have 4-times that, namely 128-bit addresses
- Digital devices will have their own IP address
  - Vint Cerf (one of Internet founders) says that every light bulb in a house will have its own IP address, for individual control
- Christian Huitema (former Chair of IETF Architecture Board) estimates that there will be 1564 addresses for each square meter of the earth’s surface!

IPv6 (cont.)

- IPv6 is designed to run well on ATM networks, while still providing efficiency for low-speed networks (e.g. wireless)
- IPv6 has various addressing mechanisms supporting plug-and-play device installation
  - good for nomadic communications (wireless, infrared)
- IPv6 multicast addresses will allow more users to be reached on a multicast
- IPv6 provides packet priorities according to time-sensitive data

IPv6 - RFC’s

- 1752 - The Recommendation for the IP Next Generation Protocol - Jan 95
- 1809 - Using the Flow Label in IPv6 - Jun 95
- 1881 - IPv6 Address Allocation Management - Dec 95
- 1883 - Internet Protocol, Version 6 Specification - Dec 95
- 1884 - IP Version 6 Addressing Architecture - Dec 95
- 1885 - Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification - Dec 95
- 1886 - DNS Extensions to Support IP Version 6 - Dec 95
- 1887 - An Architecture for IPv6 Unicast Address Allocation - Dec 95
- 1897 - IPv6 Testing Address Allocation - Jan 96
- 1924 - A Compact Representation of IPv6 Addresses - Apr 96
- 1933 - Transition Mechanisms for IPv6 Hosts and Routers - Apr 96

IPv6 : General Characteristics

- Address uses 128 bits
- It provides compatibility with IPv4
- Simple header, allowing faster routing
- New address classes with more flexibility
- Support for authentication and privacy mechanisms
- Specification of source type, that makes possible a more appropriate handing through the transmissions

Notations of IPv6 Addresses

128 bit is represented as:

- 8 integers (16-bit) separated by colons
  - each integer is represented by 4 hex digits

Example:

- FEDC:BA98:7654:3210:FEDC:BA98:7664:3210

Simplifications:

- Skip leading zeros
  - Example:1080:0000:0000:0000:0008:0800:200C:417A
  - is reduced to: 1080:0:0:0:8:800:200C:417A
- A set of consecutive nulls is replaced by ::
  - at most one :: inside an address
  - the above address is reduced to:
    - 1080::8:800:200C:417A

IPv4 compatability

- “Preferred” form:
  - ❖2001:0:FF:0:8:80:209C:5656
- Compressed form:
  - ❖FF01:0:0:0:0:0:0:43 becomes
    - FF01::43
- IPv4-compatible:
- ::ffff:c000:280 would typically be notated ::ffff:192.0.2.128 instead.
  - ❖193.140.83.27 becomes
    - Change the decimal numbers to hex numbers in order to have it correct reppresentation.
    - 0000:0000:0000:0000:0193:0140:0083:0027
    - .
    - ::193.140.83.27

IPv6 Header

0 4 12 16 24 31

Version	Trafic Class	Flow Label	
Payload Length		Next Header	Hop Limit
128 bit Source Address			
128 bit Destination Address			

IPv4 Header: to recall it

0 4 8 16 31

Ver	HL	Service Type	Total Length
Identifier		Flags	Fragment Offset
TTL	Protocol		Header Checksum
32 bit Source Address			
32 bit Destination Address			
Options and Padding			

▨ : This fields are not used in IPv6

IPv6 Packet Format

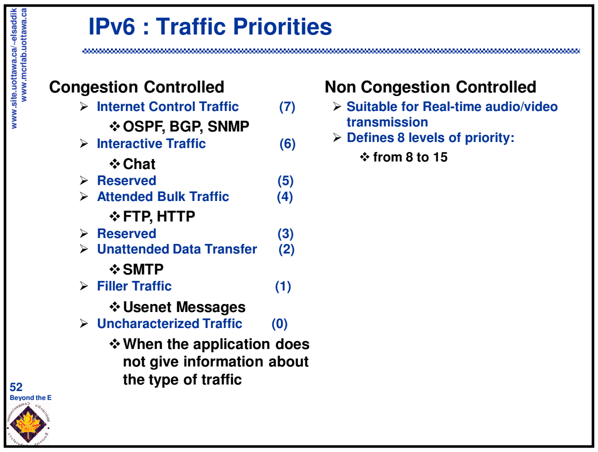
- **Version (4 bits):**
  - IP version number; the value is 6.
- **Priority (4 bits):** New Field
  - Priority value
    - to facilitate the handling of real time traffic
- **Flow label (24 bits):** New Field
  - May be used by a host to label those packets for which it is requesting special handling by routers within a network.
    - to facilitate the handling of real time traffic
- **Payload length (16 bits):** renamed (Total Length)
  - Length of the remainder of the IPv6 packet following the header; not include header length
  - max length 64Kbytes with provision for larger packets using "jumbo gram" option

IPv6 Packet Format

- **Next header (8 bits):** renamed (Protocol)
  - Identifies the type of header immediately following the IPv6 header.
- **Hop limit (8 bits):** renamed (Time to Live)
  - The remaining number of allowable hops for this packet.
- **Source address (128 bits):**
  - Originator of packet's address.
- **Destination address (128 bits):**
  - The address of the intended recipient of the packet.

IPv6 : Traffic Priorities

- 8-bit priority value
- This field may be used by a source to identify the delivery priority of its packets, relative to other packets from the same source.
- For congestion-controlled traffic, the priority values are from 0 to 7
- Higher values are used for non-congestion-controlled traffic, e.g. video.



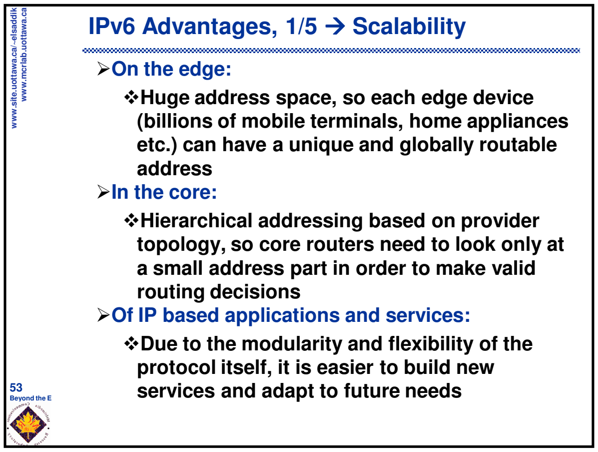
IPv6 : Traffic Priorities

Congestion Controlled

- Internet Control Traffic (7)
  - ❖ OSPF, BGP, SNMP
- Interactive Traffic (6)
  - ❖ Chat
- Reserved (5)
- Attended Bulk Traffic (4)
  - ❖ FTP, HTTP
- Reserved (3)
- Unattended Data Transfer (2)
  - ❖ SMTP
- Filler Traffic (1)
  - ❖ Usenet Messages
- Uncharacterized Traffic (0)
  - ❖ When the application does not give information about the type of traffic

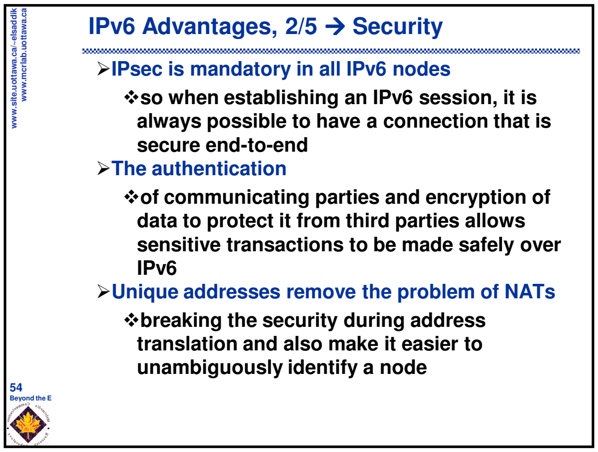
Non Congestion Controlled

- Suitable for Real-time audio/video transmission
- Defines 8 levels of priority:
  - ❖ from 8 to 15



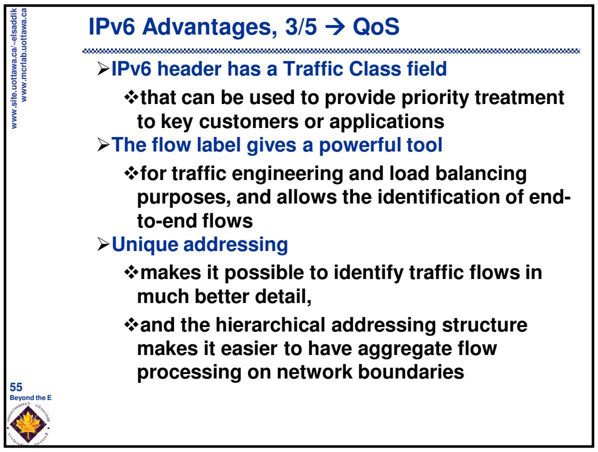
IPv6 Advantages, 1/5 → Scalability

- On the edge:
  - ❖ Huge address space, so each edge device (billions of mobile terminals, home appliances etc.) can have a unique and globally routable address
- In the core:
  - ❖ Hierarchical addressing based on provider topology, so core routers need to look only at a small address part in order to make valid routing decisions
- Of IP based applications and services:
  - ❖ Due to the modularity and flexibility of the protocol itself, it is easier to build new services and adapt to future needs



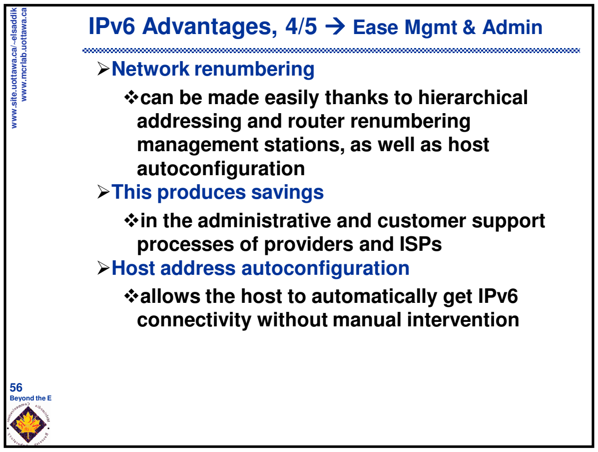
IPv6 Advantages, 2/5 → Security

- IPsec is mandatory in all IPv6 nodes
  - ❖ so when establishing an IPv6 session, it is always possible to have a connection that is secure end-to-end
- The authentication
  - ❖ of communicating parties and encryption of data to protect it from third parties allows sensitive transactions to be made safely over IPv6
- Unique addresses remove the problem of NATs
  - ❖ breaking the security during address translation and also make it easier to unambiguously identify a node



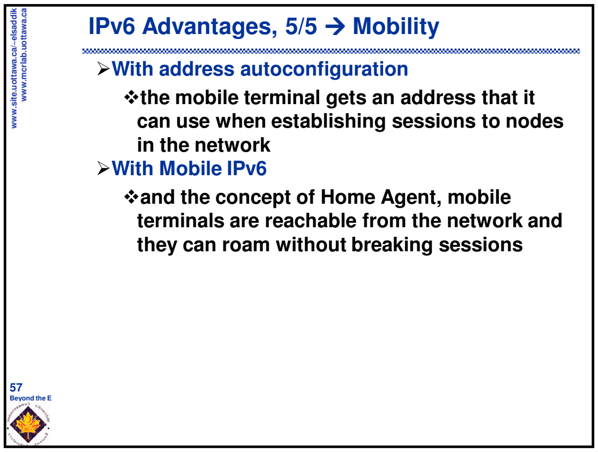
IPv6 Advantages, 3/5 → QoS

- IPv6 header has a Traffic Class field
  - that can be used to provide priority treatment to key customers or applications
- The flow label gives a powerful tool
  - for traffic engineering and load balancing purposes, and allows the identification of end-to-end flows
- Unique addressing
  - makes it possible to identify traffic flows in much better detail,
  - and the hierarchical addressing structure makes it easier to have aggregate flow processing on network boundaries



IPv6 Advantages, 4/5 → Ease Mgmt & Admin

- Network renumbering
  - can be made easily thanks to hierarchical addressing and router renumbering management stations, as well as host autoconfiguration
- This produces savings
  - in the administrative and customer support processes of providers and ISPs
- Host address autoconfiguration
  - allows the host to automatically get IPv6 connectivity without manual intervention

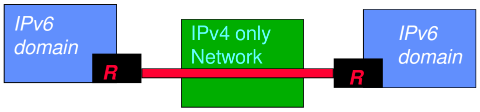


IPv6 Advantages, 5/5 → Mobility

- With address autoconfiguration
  - the mobile terminal gets an address that it can use when establishing sessions to nodes in the network
- With Mobile IPv6
  - and the concept of Home Agent, mobile terminals are reachable from the network and they can roam without breaking sessions

IP interconnectivity with underlying networks

- At the beginning, all **IPv6-capable** hosts will also be **IPv4-capable** so as to retain ***connectivity*** with the existing Internet → (Tunneling)
- To transform IPv4 into a ***dual-stack*** IPv6-capable host, it should include:
  - The IPv6 *basic code*
  - Handling IPv6 within *TCP & UDP*
  - Modify *socket interface* to support new addresses
  - Handling the interface with the *name service*



IP interconnectivity with underlying networks

- IP over Ethernet
  - IP datagrams are encapsulated in properly addressed Ethernet packets (Ethernet uses 48-bit addressing)
    - IP router uses the Address Resolution Protocol (ARP) to broadcast and find the Ethernet address of destination on the LAN (from the IP address of the destination host)
    - Reverse ARP can be used to find IP address that is bound to an Ethernet address on the LAN
- IP over ATM
  - routers need to encapsulate IP datagrams in properly addressed cells, and vice versa
    - ARP, and reverse ARP, used to do the mapping between IP address and VPI/VCI cell address of destination

Between Scylla and Charybdis

Definition: Documents

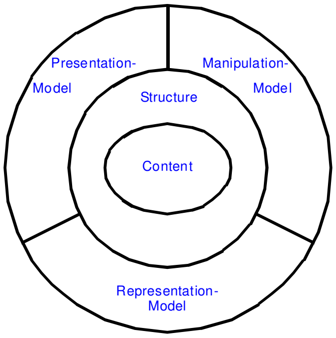
- ➢Document is a set of structured information:
  - ❖prerecorded or
  - ❖generated at presentation time
- ➢covering different media
- ➢intended for human perception
- ➢accessible to processing by a computer

Definition: Multimedia Document

- ➢Document characterized by at least:
  - ❖one continuous (time dependent) and
  - ❖one discrete (time independent) media
- ➢Integration determined by close relationship between components that are handled in different manner
- ➢Basic idea:
  - ❖Abstraction from different views and means of manipulation
  - ❖allows for integrated and uniform description and handling

---

Document Architecture



Examples:

- ➢Standard Generalized Markup Language (SGML)

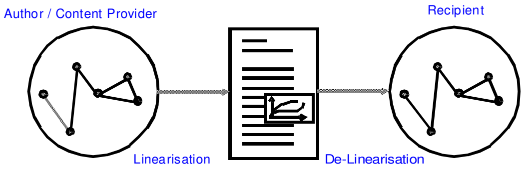
Hypertext

- Usually the transfer format (e.g. a classic book) is *linear*
- but we don't think that way
  - ❖ associations,
  - ❖ hierarchies,
  - ❖ trees,
  - ❖ semantical networks)

Conclusion:

- describe information structure in an appropriate way
- find adequate means of presentation
- transmit / exchange
  - ❖content
  - ❖structure

Hypertext



Hypertext

Process somewhat different from traditional authoring / publishing

- Author associates certain navigation structure within content
- Linearization before information exchange
- Recipient gets impression of intended structure and may navigate individually

Non-linear linkage of information:

- More than only one sequential order to retrieve information
- Reader determines 'reading path'

Hypertext

Association through references (links):
- Reference
  - contains information
  - connects pieces of information
- Producer determines the references
- Reader selects references

Content can / must be separated from representation
- Corporate identity
- limited storage or transfer bandwidth
- limited presentation support

Hypertext: Application Areas

Computer applications:
- Online documentation
- Extended help function
- World Wide Web (WWW)

Business Applications:
- Repairing and operation instructions
- Exhibition and product catalogues, advertisement

'Intellectual' Applications:
- Organization of ideas and brainstorming

General:
- Whenever the relation of information may be non-sequential

Concepts in a nutshell

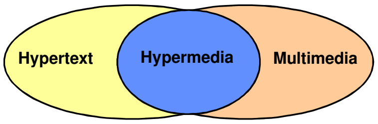
- Regular Text vs. Hypertext
  - **Sequential vs. non-sequential**
  - **Interface: a book vs. a computer system environment**
  - **Traditional author vs. "reader can be author"**
- Essential Concepts of Hypertext/Hypermedia
  - **Nodes – Units of Information**
  - **Links – Labels connecting nodes**
  - **Navigation – Process of moving through the hypertext database**
- World Wide Web (WWW, W3, „The Web"):
  - **Framework for hyperlinked documents**
  - **Huge collection of documents spread over the whole Internet**
  - **The layman's opinion: „WWW is the Internet"**

Hypermedia

Hypermedia =

Hypertext +

Multimedia



Standard Generalized Markup Language (SGML)

- ➢ Evolved from the IBM internal GML project
- ➢ Strongly supported by American publishers:
  - ❖ Authors define titles, tables in uniform way
  - ❖ Publisher determines layout
- ➢ Basic ideas:
  - ❖ Author uses tags to mark parts of the text
  - ❖ SGML determines how tags have to look like
  - ❖ User groups agree on the meaning of the tags
  - ❖ Formatter generates document layout from tags
- ➢ SGML defines:
  - ❖ Syntax
  - ❖ Not semantics

Standard Generalized Markup Language (SGML)

Relationship between:

- ➢ Document
  - ❖ data content
  - ❖ markups
- ➢ Document Type Definition (DTD):
  - ❖ set of markup declarations
  - ❖ define
    - element types
    - attributes of elements
    - hierarchical relationships between elements
- ➢ Procedures
  - ❖ specify the document processing
  - ❖ correspond to functions of the formatter

```
<title> MM Communications
</title>
<author> Prof
</author>
<Summary> Multimedia,
Communications, protocols
</summary>
…….
```

SGML: Markup Categories

Descriptive Markup (Tags):

- Defines the structure of the document
- Form: <tag> text </tag>
- Example: <paragraph> This is a paragraph ... </paragraph>

Entity Reference:

- For symbol substitution and file, data set, and variable imbedding
- Example in German: &Ae.rger ... means ... Ärger

SGML: Markup Categories

Markup Declaration:

- Can define the entities referred to by the Entity References:
  - example: **<!ELEMENT Ae (...)>**
- Can define rules for the structure of the document:
  - example:

```
<!Element paper (preamble, body, postamble)>
<!Element preamble (title, author, side)>
<!Element title (#CDATA)> --character data
<!Element body (...)>
...
```

Processing instructions:

- Instructions used by other programs, e.g. a formatter
- E.g. for the imbedding of different media

SGML and Multimedia

Imbedding of Multimedia Data:

- As external document parts from separate files
- Originally embedded as CGM (Computer Graphics Metafile) graphic
  - concrete data referred by using NDATA
- Now: HTML, HyTime

Example:

```
<!ATTLIST video id ID #IMPLIED>
<!ATTLIST video synch #IMPLIED>
<!ELEMENT video (audio, movpic)>
<!ELEMENT audio (#NDATA)> -- non-text media
<!ELEMENT movpic (#NDATA)> -- non-text media
...
<!ELEMENT story (preamble, body, postamble)>
```

SGML Conformant Languages

HTML

- ➢ Hypertext Markup Language
- ➢ For description of hypertext / hypermedia in the World Wide Web (WWW)

HyTime

- ➢ ISO standard for structured presentation of hypermedia information

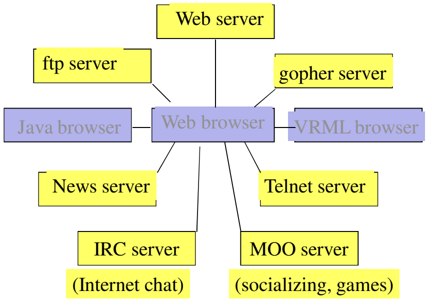
SMDL

- ➢ Standard Music Description Language

XML

- ➢ Extensible Markup Language
- ➢ driven by W3C consortium (XML working group)
- ➢ designed to enable the use of SGML on the WWW
- ➢ note:
  - ❖metalanguage
  - ❖lets you design your own markup language
- ➢ its: SGML-- rather than HTML++

Web browsers



Web server

ftp server

gopher server

Java browser

Web browser

VRML browser

News server

Telnet server

IRC server

(Internet chat)

MOO server

(socializing, games)

The Web: the http protocol

http: hypertext transfer protocol

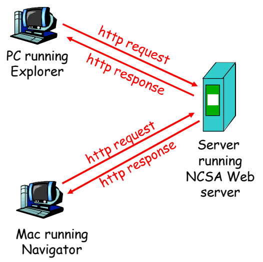
Web's application layer protocol

client/server model

- ➢ ***client:*** browser that requests, receives, "displays" Web objects
- ➢ ***server:*** Web server sends objects in response to requests

http1.0: RFC 1945

http1.1: RFC 2068



The http protocol: more

http: TCP transport service:

- client initiates TCP connection (creates socket) to server, port 80
- server accepts TCP connection from client
- http messages (application-layer protocol messages) exchanged between browser (http client) and Web server (http server)
- TCP connection closed

http is "stateless"

- server maintains no information about past client requests

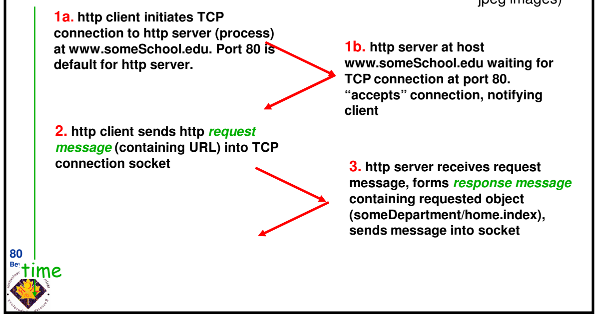
aside

Protocols that maintain "state" are complex!

past history (state) must be maintained

if server/client crashes, their views of "state" may be inconsistent, must be reconciled

http example



Suppose user enters URL
www.someSchool.edu/someDepartment/home.index

(contains text, references to 10 jpeg images)

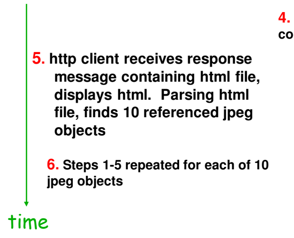
1a. http client initiates TCP connection to http server (process) at www.someSchool.edu. Port 80 is default for http server.

1b. http server at host www.someSchool.edu waiting for TCP connection at port 80. "accepts" connection, notifying client

2. http client sends http *request message* (containing URL) into TCP connection socket

3. http server receives request message, forms *response message* containing requested object (someDepartment/home.index), sends message into socket

http example (cont.)



4. http server closes TCP connection.

5. http client receives response message containing html file, displays html. Parsing html file, finds 10 referenced jpeg objects

6. Steps 1-5 repeated for each of 10 jpeg objects

Non-persistent and persistent connections

Non-persistent

HTTP/1.0

- server parses request, responds, and closes TCP connection
- 2 RTTs to fetch each object
- Each object transfer suffers from slow start

Persistent

default for HTTP/1.1

- on same TCP connection: server, parses request, responds, parses new request,..
- Client sends requests for all referenced objects as soon as it receives base HTML.
- Fewer RTTs and less slow start.

But most 1.0 browsers use parallel TCP connections.

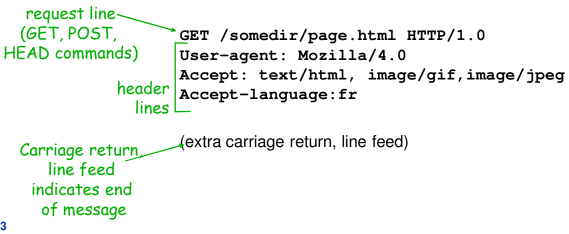
---

http message format: request

two types of http messages: ***request, response***

http request message:

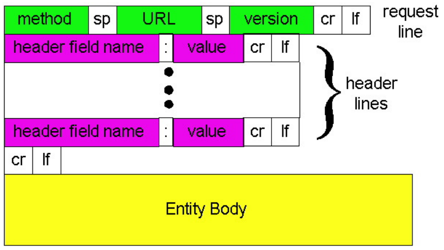
- **ASCII (human-readable format)**



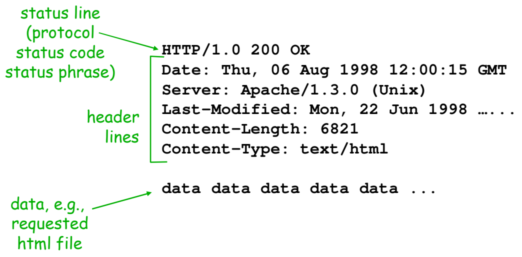
```
GET /somedir/page.html HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif,image/jpeg
Accept-language:fr

(extra carriage return, line feed)
```

---

http request message: general format



http message format: respone



```
HTTP/1.0 200 OK
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 …...
Content-Length: 6821
Content-Type: text/html

data data data data data ...
```

---

http response status codes

In first line in server->client response message.
A few sample codes:

- `200 OK`
  - request succeeded, requested object later in this message
- `301 Moved Permanently`
  - requested object moved, new location specified later in this message (Location:)
- `400 Bad Request`
  - request message not understood by server
- `404 Not Found`
  - requested document not found on this server
- `505 HTTP Version Not Supported`

---

http request methods

- HTTP allows an open-ended set of methods that indicate the purpose of a request:
  - GET:
    - read whatever data identfied bei the requested URI
  - HEAD:
    - read whatever data's header (containing info about data)
  - PUT:
    - store whatever data under the requested URI
  - POST:
    - append data under the supplied request URI
  - DELETE:
    - remove data identified by the requested URI
  - LINK:
    - connect two resources
  - UNLINK:
    - break existing connection
  - .....

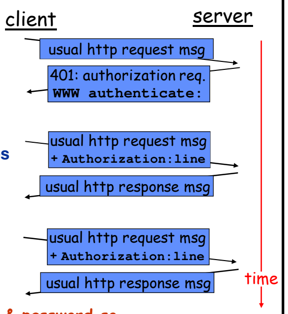
User-server interaction: authentication

Authentication goal: control access to server documents

- **stateless:** client must present authorization in each request
- authorization: typically name, password
  - `authorization:` header line in request
  - if no authorization presented, server refuses access, sends

    `WWW authenticate:`

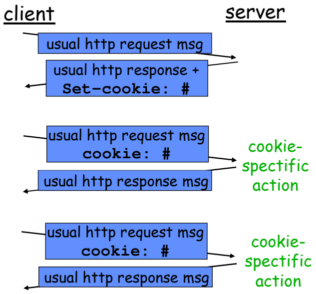
    header line in response



Browser caches name & password so that user does not have to repeatedly enter it.

User-server interaction: cookies

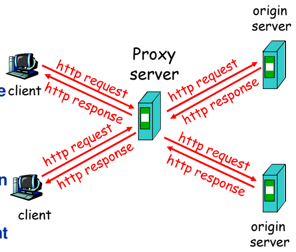
- server sends "cookie" to client in response mst
  `Set-cookie: 8783294`
- client presents cookie in later requests
  `cookie: 8783294`
- server matches presented-cookie with server-stored info
  - authentication
  - remembering user preferences, previous choices

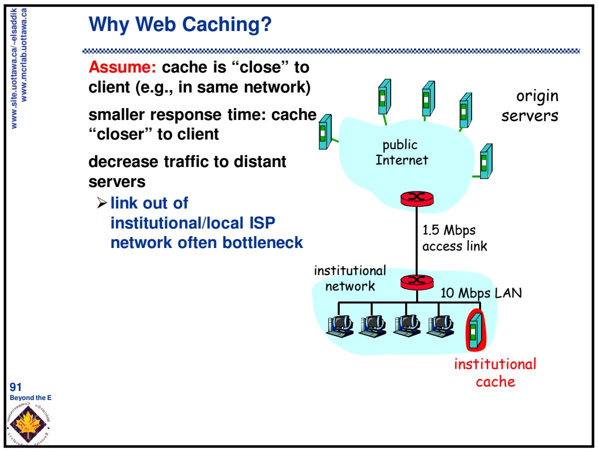


Web Caches (proxy server)

Goal: satisfy client request without involving origin server

- user sets browser: Web accesses via web cache
- client sends all http requests to web cache
  - if object at web cache, web cache immediately returns object in http response
  - else requests object from origin server, then returns http response to client





Why Web Caching?

Assume: cache is "close" to client (e.g., in same network)

smaller response time: cache "closer" to client

decrease traffic to distant servers

- link out of institutional/local ISP network often bottleneck

HTML Limitations

HTML is limited as a multimedia specification language

- does not generally support dynamic presentations
- does not offer much support for audio
- does not describe how to render any given data format , thus each browser must be equipped to handle data format

WWW browsers fill needs by:

- plugins, helper applications , applets, scripting languages

Delivery techniques and Protocols

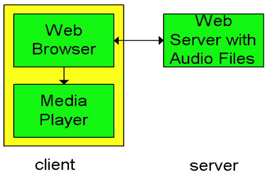
- MIME was the first widely used multimedia transfer mechanism on the Internet
- HTTP protocol:
  - optimized the transfer of text and allowed communication with servers and other protocols, such as FTM, SNMP, WAIS , Gopher
- Long latency for audio and video transfers
- New continuous stream protocols developed
  - RealAudio: bypasses HTTP server, connecting directly to a dedicated server
- RSVP, RTP, MBONE, RTSP

Streaming

- **Important and growing application due to reduction of storage costs, increase in high speed net access from homes, enhancements to caching and introduction of QoS in IP networks**
- **Audio/Video file is segmented and sent over either TCP or UDP, public protocol:** Real-Time Protocol (RTP)
- **User interactive control is provided, e.g. the public protocol** Real Time Streaming Protocol (RTSP)
- Helper Application:
  - **displays content, which is typically requested via a Web browser; e.g. RealPlayer; typical functions:**
    - **Decompression**
    - **Jitter removal**
    - **Error correction:**
      - use redundant packets to be used for reconstruction of original stream
    - **GUI for user control**

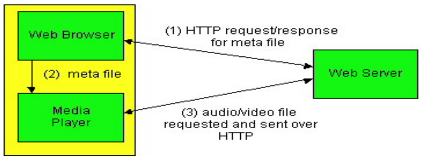
Streaming From Web Servers

- **Audio: in files sent as HTTP objects**
- **Video (interleaved audio and images in one file, or two separate files and client synchronizes the display) sent as HTTP object(s)**
- **A simple architecture is to have the Browser requests the object(s) And after their reception pass them to the player for display**
- **No pipelining**



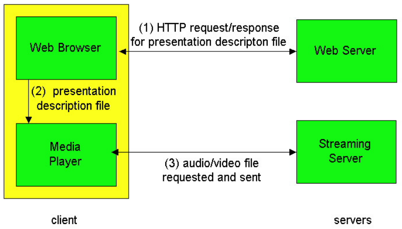
Alternative

- **set up connection between server and player, then download**
- **Web browser requests and receives a** Meta File **(a file describing the object) instead of receiving the file itself;**
- **Browser launches the appropriate Player and passes it the Meta File;**
- **Player sets up a TCP connection with Web Server and downloads the file**



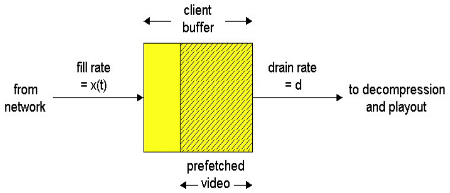
Using a Streaming Server

- This gets us around HTTP, allows a choice of UDP vs. TCP and the application layer protocol can be better tailored to Streaming; many enhancements options are possible (see next slide)



Options When Using a Streaming Server

- **Use UDP**
  - Server sends at a rate (Compression and Transmission) appropriate for client; to reduce jitter, Player buffers initially for 2-5 seconds, then starts display
- **Use TCP**
  - sender sends at maximum possible rate under TCP;
  - retransmit when error is encountered; Player uses a much large buffer to smooth delivery rate of TCP



Options When Using a Streaming Server

Transport
Protocol
Solution

RTP: Real-time Transport Protocol

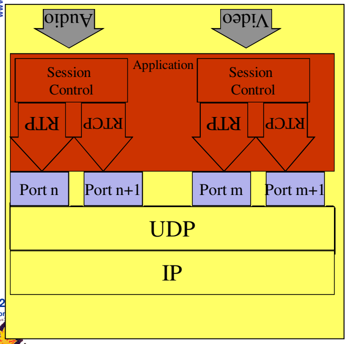
- IETF Audio/Video Transport WG
  - RTPv1 RFC 1889 (January 1996)
  - RTPv2 draft-ietf-avt-rtp-new-09.txt (March 2001)
- RTP is an end-to-end network transport functions for applications transmitting real-time data such as audio,video.
  - including timing reconstruction, loss detection, security and content identification.
- While UDP/IP is its initial target networking environment, efforts have been made to make RTP transport-independent
  - Normally used over UDP and uses UDP services such as checksums and multiplexing.
- Does not provide a mechanism for timely delivery of media/ No QOS guarantees
- comes with a control protocol (RTCP) which is used to monitor data delivery and to provide information about participants in the conference
  - its companion control protocol
  - does not guaranty anything either!

RTP -- Design goals:

- Flexible
  - provide mechanisms, do not dictate algorithms!

  ⇒ instantiations for H261, MPEG1/2/...
- protocol neutral
  - UDP/IP, ATM
  - private networks...
- Scalable
  - unicast, multicast, from 2 to ∞
- separate control/data
  - some functions may be taken over by conference control protocol (e.g. RTSP)

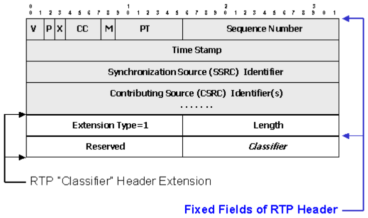
RTP in a video/audio conference



RTP
- for data
- Important fields in header
- Sequence number
- timestamp
- synchronization ID

RTCP
- for QoS monitoring and control
- provides feedback on the quality of the data distribution

RTP Header Format



- V indicates the version of RTP used
- P indicates the padding, a byte not used in the bottom packet to reach the parity packet dimension
- X is the presence of the header extension
- CC field is the number of CSRC identifiers following the fixed header. CSRC field are used, for example, in conference case
- M is a marker bit
- PT is the payload typ

RTP

- **standard packet structure : includes fields for audio/video type, sequence number, timestamp, and more.**
- **used for transporting common formats (WAV, GSM, MPEG1, MPEG2)**
- **runs on top of UDP: code in the application encapsulates chunks of data in RTP packets, and sends RTP packets to a UDP socket interface, where it is encapsulated in a UDP packet.**

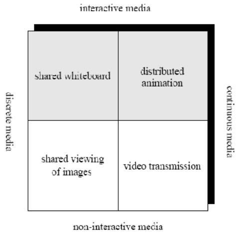
- **As currently defined, RTP does not provide any mechanisms for recovering for packet loss. RTP probably has the necessary header information (like sequence numbers) for some forms of error recovery by retransmission.**
- **RTP can be used on IPv6**
- **RTP contains no specific assumptions about the capabilities of the lower layers, except that they provide framing. It contains no network-layer addresses, so that RTP is not affected by addressing changes.**
- **RTP can be used over ATM and on asymmetric networks (ADSL, cable modems, satellite networks)**

RTP

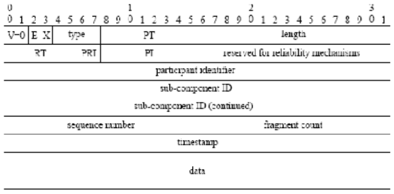
- Uses a special set of messages (RTCP) to exchange periodic reports
- One RTP session, one media flow
- Payload Type: 7 bits, providing 128 possible different types of encoding; eg PCM, MPEG2 video, etc.
  - different media are not multiplexed
- Sequence Number: 16 bits; random number incremented by one for each RTP data packet sent; used to detect packet loss
- Timestamp: 32 bytes; gives the sampling instant of the first audio/video byte in the packet; used to remove jitter introduced by the network
  - clock frequency depends on applications
  - random initial value
  - several packets may have equal timestamps (eg. same video frame), or even in disorder (eg. interpolated frames in MPEG)
- Synchronization Source identifier (SSRC): 32 bits; an id for the source of a stream; assigned randomly by the source
- Miscellaneous fields: Contributing Source identifier (CSRC)

Real-Time Protocol for Interactive applications (RTP-I)

- Application Layer Protocol
- Designed for network distributed interactive applications
- Framing of events and state data
- Uses the concept of Event Packet Type (Event, State, Delta_State, State_Query)



RTP-I Header Format



- o V indicates the version of RTP-I used
- o E and X fields are used for fragmentation and reassembly
- o type identifies the contents of the packet (Event, State, Delta_State, State_Query)
- o RT defines the reliability mechanism
- o Timestamp – at which the event has happened

Real Time Streaming Protocol (RTSP)

- ➢ **For user to control display: rewind, fast forward, pause, resume, etc…**
- ➢ **Application level** framework **for control**
  - ❖**not really a protocol**
- ➢ **Sends requests and responses**
  - ❖does NOT carry data
- ➢ **Designed to** work on top of RTP
  - ❖**underlying protocols can be either UDP or TCP**
- ➢ **Defines standard** methods **for real-time media connection set-up and transmission control**
- ➢ **Out-of-band protocol**
  - ❖**uses two connections:**
    - **one for control messages (Port 554) and**
    - **for media stream**

RTSP generalities

Protocol design

- text-based protocol
- transport protocol independent
- supports any session description (sdp, xml, etc.)
- A meta file is communicated to web browser which then launches the Player;
  - Player sets up an RTSP connection for control messages in addition to the connection for the streaming media
- HTTP like protocol (remote control over the network) for controlling the delivery of real-time interactive and stored clips.
- Used in popular streaming applications like RealPlayer, QuickTime.
  - Realplayer uses RTP-like (but proprietary) transport protocol for carrying the media
  - QuickTime uses RTP

RTSP methods

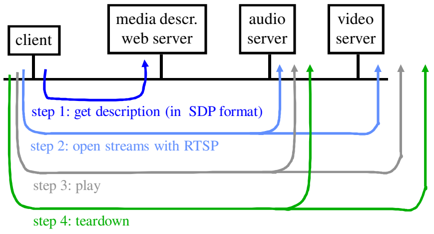
Major methods

- SETUP:
  - server allocates resources for a stream and starts an RTSP session
- PLAY:
  - starts data tx on a stream
- PAUSE:
  - temporarily halts a stream
- TEARDOWN:
  - free resources of the stream, no RTSP session on server any more

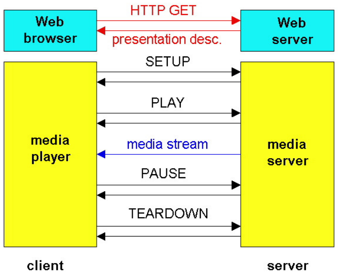
Additional methods

OPTIONS:	get available methods
ANNOUNCE:	change description of media object
DESCRIBE:	get low level descr. of media object
RECORD:	server starts recording a stream
REDIRECT:	redirect client to new server
SET_PARAMETER:	device or encoding control

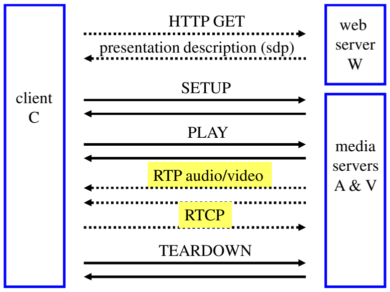
Example: media on demand, unicast



Another View



3rd View



Meta File Example

```
<title>Twister</title>
<session>
      <group language=en lipsync>
            <switch>
            • <track type=audio
            •                    e="DVI4/16000/2" pt="90 DVI4/8000/1"
            •
              src="rtsp://audio.example.com/twister/audio.en/hifi">

            •               <track type=audio
            •                    e="PCMU/8000/1"
            •                    src =
              "rtsp://audio.example.com/twister/audio.en/lofi">
         </switch>
        <track type="video/jpeg"
                  src="rtsp://video.example.com/twister/video">
      </group>
</session>
```

RTSP Exchange Example

```
C: SETUP rtsp://audio.example.com/twister/audio RTSP/1.0
      Transport: rtp/udp; compression; port=3056; mode=PLAY

S: RTSP/1.0 200 1 OK
       Session 4231

C: PLAY rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
       Session: 4231
       Range: npt=0-

C: PAUSE rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
       Session: 4231
       Range: npt=37

C: TEARDOWN rtsp://audio.example.com/twister/audio.en/lofi RTSP/1.0
       Session: 4231

S: 200 3 OK
```

Recent Multimedia Web developments

- Macromedia Flash
- XML Family (DHTML, SMIL, etc…)
- Virtual Reality Modeling Language (VRML)
  - 3D objects manipulation, Virtual Worlds
- JAVA language from Sun Microsystems
  - multimedia scripts
- Audio / Video streaming
- WebTV
- HAPTIC environments