Network architectures, protocols, and mechanisms to support multimedia

Jakub Jakubiak
SGN-5876 Energy Efficient Scalable Multimedia Coding and Communications
Department of Communications Engineering
Outline

1. Network basics (refreshment)
2. OSI reference and TCP/IP network models
3. Physical/data link layer (wireless)
4. IP protocol
5. Transport protocols (TCP and UDP)
6. Application layer: Real-time Transport (RTP) and Real-time Transport Control Protocol (RTCP)

Slides partially sourced from W. Stallings „Data and Computer Communications”
NETWORK BASICS
A Communications Model

Source
  • generates data to be transmitted

Transmitter
  • Converts data into transmittable signals

Transmission System
  • Carries data

Receiver
  • Converts received signal into data

Destination
  • Takes incoming data
A Communications Model

(a) General block diagram

(b) Example
A Communications Model

1. Input information $m$
2. Input data $g(t)$
3. Transmitted signal $s(t)$
4. Received signal $r(t)$
5. Output data $g'(t)$
6. Output information $m'$
Key Communications Tasks

Transmission System Utilization
Interfacing
Signal Generation
Synchronization
Exchange Management
Error detection and correction
Addressing and routing
Recovery
Message formatting
Security
Network Management
Networking

Point to point communication not usually practical

- Devices are too far apart
- Large set of devices would need impractical number of connections

A communication network solves that problem
Simplified Network Model
A protocol is used for communications between entities in different systems

- Entity – user application SW, file transfer packages, etc.
- System – computers, terminals, etc.

In general

- An entity is anything capable of sending or receiving data
- A system is a physically distinct object that contains one or more entities

Entities must speak the same language, so the protocols on each side must understand each other
Protocol characteristics

Network protocol classification
• Direct or indirect
• Monolithic or structured
• Symmetric or asymmetric
• Standard or nonstandard
Direct or Indirect

Direct communication

• Systems with a point to point link: data and control information pass directly between entities
• Systems share a multi-point link: the entities must deal with the issue of access control
• Data passes without intermediate active agent

Indirect communication

• Data transfer depends on other entities
• Switched networks, internetworks or internets
Monolithic or Structured

Communications is a complex task
• Too complex for a single entity

Structured design breaks down the problem
Layered structure

• Usage of structured design / implementation techniques
  • Instead of a single protocol use a set of protocols
  • Hierarchical or layered structure
Symmetric or Asymmetric

Symmetric
- Communication between peer entities
- Both ends behave in the same way
- Peer to peer architecture

Asymmetric
- May be dedicated by the logic of an exchange or by desire to keep one of the entities or systems as simple as possible
- Client/server architecture
Standard or Nonstandard

- Nonstandard protocols built for specific systems and tasks
  - K sources and L destinations lead to $K \times L$ protocols and $2^*K^*L$ implementations are required

- Common protocol
  - $K+L$ implementations needed

(a) Without standards: 12 different protocols; 24 protocol implementations

(a) With standards: 1 protocol; 7 implementations
Network Protocols - Layering

High degree of cooperation between hosts requires high complexity
  • Special logic has to be implemented
Instead of implementing all in a single module, break the task into separately implemented subtasks
  • Gather the tasks into several groups
  • Each subtask is simple, easy to solve independently
  • Groups identified as ‘layers’, their tasks performed by a corresponding protocol
E.g. a simple file transfer system could use three modules
  • File transfer application
  • Communication service module
  • Network access module
Layered approach!
Example: File Transfer Architecture

- **File transfer module**: two modules on two systems exchange files and commands
- **Communication service module**: responsible for making sure that the file transfer commands and data are reliably exchanged
- **Network access module**: interacts with the network

![Diagram of file transfer architecture]
Network Protocols - Functions

- Encapsulation
- Segmentation and reassembly
- Connection control
- Ordered delivery
- Flow control
- Error control
- Addressing
- Multiplexing
- Transmission services

NOTE: note all protocols have all functions!
ISO/OSI AND TCP/IP
OSI Model

- Open Systems Interconnection (OSI) developed by International Organization for Standardization (ISO)
- 7 layers
- A conceptual and functional framework
- Not specified in OSI:
  - Services or protocols
  - Implementation specification for systems
  - Basis for evaluating the conformance of implementations
- A theoretical system that was delivered too late
- TCP/IP is the de facto standard
OSI Layers

Application process

- Application layer (Layer 7)
- Presentation layer (Layer 6)
- Session layer (Layer 5)
- Transport layer (Layer 4)
- Network layer (Layer 3)
- Data link layer (Layer 2)
- Physical layer (Layer 1)

Peer interfaces

- Application layer (Layer 7)
- Presentation layer (Layer 6)
- Session layer (Layer 5)
- Transport layer (Layer 4)
- Network layer (Layer 3)
- Data link layer (Layer 2)
- Physical layer (Layer 1)

Application process

Physical media for OSI ("Layer 0")
<table>
<thead>
<tr>
<th>OSI model</th>
<th>OSI protocol suite (example)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application layer</td>
<td>Message Handling System / ITU-T X.400-series</td>
</tr>
<tr>
<td>(Layer 7)</td>
<td></td>
</tr>
<tr>
<td>Presentation layer</td>
<td>Presentation Protocol (PP) / ISO 8823, ITU-T X.226</td>
</tr>
<tr>
<td>(Layer 6)</td>
<td></td>
</tr>
<tr>
<td>Session layer</td>
<td>Session Protocol (SP) / ISO 8327, ITU-T X.225</td>
</tr>
<tr>
<td>(Layer 5)</td>
<td></td>
</tr>
<tr>
<td>Transport layer</td>
<td>Transport Protocol Class 4 (TP4) / ISO 8073</td>
</tr>
<tr>
<td>(Layer 4)</td>
<td></td>
</tr>
<tr>
<td>Network layer</td>
<td>Packet-Layer Protocol (PLP) / X.25</td>
</tr>
<tr>
<td>(Layer 3)</td>
<td></td>
</tr>
<tr>
<td>Data link layer</td>
<td>Link Access Procedure, Balanced (LAPB) / X.25</td>
</tr>
<tr>
<td>(Layer 2)</td>
<td></td>
</tr>
<tr>
<td>Physical layer</td>
<td>Physical layer synchronous serial protocol (X.21bis) / X.25</td>
</tr>
<tr>
<td>(Layer 1)</td>
<td></td>
</tr>
</tbody>
</table>
Basic Design Principles

• The layers use control information to communicate with peer layers
• The control information consists of specific requests and instructions exchanged between peer OSI entities
• Control information typically takes form of headers and/or trailers
  • **Headers** are *prepended* to data that has been passed down from upper layers
  • **Trailers** are *appended* to that data
OSI Message Receiving

Actual transmission of signals over physical media

Application process
- Application layer (Layer 7)
- Presentation layer (Layer 6)
- Session layer (Layer 5)
- Transport layer (Layer 4)
- Network layer (Layer 3)
- Data link layer (Layer 2)
- Physical layer (Layer 1)

Binary data: 11101010001110000110111110011010001101011
Communications with a relay system
TCP/IP Protocol Architecture

Developed by the US Defense Advanced Research Project Agency (DARPA) for its packet switched network (ARPANET)

Used by (on) the Internet, no official model exists

Layer separation is not necessarily strict

Consists of (depending on the reference one uses) 3 to 5 layers

- **Application** layer
- Host to host or **transport** layer
- **Internet** layer
- **Network access** layer
- **Physical** layer
# TCP/IP Protocol Architecture

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Five layers</strong></td>
<td><strong>Four+one layers</strong></td>
<td><strong>Five layers</strong></td>
<td><strong>Five layers</strong></td>
<td><strong>Four layers</strong></td>
<td><strong>Three layers</strong></td>
</tr>
<tr>
<td>&quot;Five-layer Internet model&quot; or &quot;TCP/IP protocol suite&quot;</td>
<td>&quot;TCP/IP 5-layer reference model&quot;</td>
<td>&quot;TCP/IP model&quot;</td>
<td>&quot;TCP/IP 5-layer reference model&quot;</td>
<td>&quot;Internet model&quot;</td>
<td>&quot;Internet model&quot;</td>
</tr>
<tr>
<td>Application</td>
<td>Application</td>
<td>Application</td>
<td>Application</td>
<td>Application</td>
<td>Application/Process</td>
</tr>
<tr>
<td>Transport</td>
<td>Transport</td>
<td>Host-to-host or transport</td>
<td>Transport</td>
<td>Transport</td>
<td>Transport</td>
</tr>
<tr>
<td>Network</td>
<td>Internet</td>
<td>Internet</td>
<td>Internet</td>
<td>Internet</td>
<td>Internetwork</td>
</tr>
<tr>
<td>Data link</td>
<td>Data link (Network interface)</td>
<td>Network access</td>
<td>Data link</td>
<td>Link</td>
<td>Network interface</td>
</tr>
<tr>
<td>Physical</td>
<td>(Hardware)</td>
<td>Physical</td>
<td>Physical</td>
<td>Physical</td>
<td></td>
</tr>
</tbody>
</table>
The layers communicate by performing encapsulation / decapsulation of PDUs

Different layers follow different conventions for naming the PDUs
TCP/IP Protocol Architecture Model

SourceSystem

Source -> Transmitter -> Transmission System -> Receiver -> Destination System

Application

TCP
IP
Network Access
Physical

Source System

Network

Application

TCP
IP
Network Access
Physical

Destination System
Physical Layer

Sometimes not included in the model (below the link layer)

• Physical interface between data transmission device (e.g. computer) and transmission medium or network

• Characteristics of transmission medium
  • Signal levels
  • Data rates

• Medium: radio wave, optical fiber, copper wire (coax or TP), IR,
Network Access Layer

• Also called the network interface layer, or the data link layer
• Exchange of data between end system and network
• Destination address provision
• Other services, e.g. priority
• The TCP/IP protocol suite defines 2 protocols at the link layer:
  • Serial Line Internet Protocol (SLIP)
  • Point-to-Point Protocol (PPP)
• Plenty of non-TCP/IP protocols, e.g. the IEEE 802 family
Internet Layer

- Sometimes called the IP layer
- Systems may be attached to different networks
- Routing functions across multiple networks
- Implemented in end systems and routers
- Protocols:
  - Internet Protocol (IP)
  - Internet Control Message Protocol (ICMP)
  - Address Resolution Protocol (ARP)
  - Reverse Address Resolution Protocol (RARP)
  - Etc.
Transport Layer

• Reliable delivery of data
• Ordering of delivery
• 2 protocols defined in the TCP/IP protocol suite:
  • Transmission Control Protocol (TCP)
  • User Datagram Protocol (UDP)
Application Layer

- Used by application processes
- Numerous protocols for end-user applications and network services:
  - HyperText Transfer Protocol (HTTP)
  - File Transfer Protocol (FTP)
  - Internet Message Access Protocol (IMAP)
  - Post Office Protocol Version 3 (POP3)
  - Domain Name Service (DNS)
  - Simple Network Management Protocol (SNMP)
  - Etc.
OSI model vs. the TCP/IP model

OSI model:
- Application layer (Layer 7)
- Presentation layer (Layer 6)
- Session layer (Layer 5)
- Transport layer (Layer 4)
- Network layer (Layer 3)
- Data link layer (Layer 2)
- Physical layer (Layer 1)

TCP/IP model:
- Application layer
- Transport layer
- Internet layer
- Link layer

TCP/IP protocol suite:
- HTTP, FTP, IMAP, SMTP, POP3, DNS, SNMP, Telnet, ...
- TCP, UDP
- IP, ICMP, ARP, RARP, ...
- SLIP, PPP
- IEEE stds,… (DL & PHY layers)
OSI model vs. the TCP/IP model

<table>
<thead>
<tr>
<th>OSI</th>
<th>TCP/IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Application</td>
</tr>
<tr>
<td>Presentation</td>
<td>Transport (host-to-host)</td>
</tr>
<tr>
<td>Session</td>
<td>Internet</td>
</tr>
<tr>
<td>Transport</td>
<td>Network Access</td>
</tr>
<tr>
<td>Network</td>
<td></td>
</tr>
<tr>
<td>Data Link</td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td>Physical</td>
</tr>
</tbody>
</table>

User Space  
Software  
Firmware  
Hardware  
Operating System
PHYSICAL AND DATA LINK LAYERS
Physical media

Transmission media include: twisted pair (TP), coaxial cable, optical fiber, radio waves, infra-red (IR)

Guided media provides a conductor from one device to another
- Example: twisted pair, coaxial cable, optical fiber

Unguided media transport electromagnetic waves without using a physical conductor
- Example: radio, infra-red
Wireless Transmission

Unguided media
Transmission and reception via antenna

- Directional
  - Focused beam
  - Careful alignment required

- Omnidirectional
  - Signal spreads in all directions
  - Can be received by many antennae
Frequencies

2GHz to 40GHz
- Microwave
- Highly directional
- Point to point
- Satellite

30MHz to 1GHz
- Omnidirectional
- Broadcast radio

3 x 10^{11} to 2 x 10^{14}
- Infrared
- Local
Radio propagation mechanisms

- **Reflection**
  - Wave hits an object which is very large compared to its wavelength
  - Phase shift of 180 degrees between the incident and the reflected rays

- **Diffraction**
  - Wave hits an object that is comparable to its wavelength;
  - Wave bends at the edges of the object, propagating in different directions

- **Scattering**
  - Wave goes through a medium with objects that are small compared to its wavelength
  - Wave gets scattered into several weaker outgoing signals
Radio propagation mechanisms
Radio propagation models

• Large-scale propagation models:
  • Predict the average received signal strength at a given distance from transmitter
  • Capture the path loss component
  • Used to estimate the radio coverage area around the transmitter

• Small-scale propagation models:
  • Characterize the rapid fluctuations of the received signal strength
  • Capture influence of multipath components
  • Used to (dynamically) evaluate performance of data transmission over the wireless channel
Radio propagation models

Small scale $\alpha(t)$ vs. large scale $m(t)$ propagation
BER in wireless

- Wireless environment is characterized by a relatively high BER
  - **BER** (Bit Error Rate) number of erroneously received bits of a data stream over the total number of received bits
  - Similarly **PER**: Packet Error Rate

- Error causes:
  - Noise
  - Interference
  - Distortion
  - Bit synchronization errors
  - Signal attenuation
  - Multipath fading etc.
INTERNET LAYER
Internet Protocol (IP)

The Internet Protocol (IP) is the heart of the TCP/IP protocol suite
- The principal communications protocol of the Internet

Specifies interfaces with higher layer protocols
- e.g. TCP, UDP

Specifies protocol format and mechanisms

The Network layer is charged with the delivery of data between devices that may be on different, interconnected networks
Internet Protocol (IP)

- Universally-addressed
- Underlying-protocol independent
- Connectionless
- Unreliable (aka best-effort)
  - No delivery guarantee
  - No Quality of Service (QoS), no acknowledgements
  - These are implemented in other layer protocols
- Thus low complexity and high flexibility
  - Power of layering!
Internet Protocol (IP)
IP Protocol v4 - format

- Version
- IHL
- Type of Service
- Total Length
  - Identification
  - Flags
  - Fragment Offset
- Time to Live
- Protocol
- Header Checksum
- Source Address
- Destination Address
- Options + Padding
Header Fields (1)

• Version
  • Currently 4
  • Slowly transiting to IP version 6
    • Different frame format

• Internet header length
  • In 32 bit words
  • Including options

• Type of service
  • Redefined as 6-bit Differentiated Services Field (DS) and a 2-bit Explicit Congestion Notification (ECN)

• Total length
  • Of packet (incl. the IP header), in octets
Header Fields (2)

- **Identification**
  - Sequence number used with addresses and user protocol to uniquely identify a packet

- **Flags (3 bits)**
  - 1st reserved (always ‘0’)
  - 2nd ‘More fragments’
  - 3rd ‘Don’t fragment’

- **Fragmentation offset**

- **Time To Live (TTL)**

- **Protocol**
  - Higher (transport) layer protocol to receive the payload at destination, e.g. TCP (0x06) or UDP (0x11)
Header Fields (3)

• Header checksum
  • Verified and recomputed at each router
  • 16 bit ones complement sum of all 16 bit words in header
  • Set to zero during calculation

• Source address

• Destination address

• Options

• Padding
  • Fill to multiple of 32 bits long words
Header Fields (4)

- Options Field
  - Security
  - Source routing
  - Route recording
  - Stream identification
  - Timestamping

- Data Field
  - Carries user data from next layer up (payload)
  - Integer multiple of 8 bits long (octet)
  - Max length of a packet (header plus data): 65,535 octets
IP Version Number

- IP v0-3 – development versions, replaced
- IP v4 – first standardized & currently version
- IP v5 – (experimental) streaming protocol
- IP v6 – successor of IP v4
  - During development it was called IP Next Generation (IPng)
IPv4 & IPv6 statistics from:
http://ipv6.he.net/ (app ByeBye v4.)

3.5, 1.5, 0.5 years ago and today
TRANSPORT LAYER
The transport layer

- Resides between the application and internet layers
- Provides end-to-end data transfer service for applications
- Uses the services from the underlying internet layer
- Hides details of underlying network from the application layer

<table>
<thead>
<tr>
<th>TCP/IP stack</th>
<th>PDU names</th>
<th>Encapsulation / Decapsulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application layer</td>
<td>Data stream / Message</td>
<td>data</td>
</tr>
<tr>
<td>Transport layer</td>
<td>TCP segment / UDP datagram</td>
<td>Tr-h data</td>
</tr>
<tr>
<td>Internet layer</td>
<td>IP packet</td>
<td>IP-h Tr-h data</td>
</tr>
<tr>
<td>Link layer</td>
<td>Link frame</td>
<td>L-header IP-h Tr-h data</td>
</tr>
<tr>
<td>Physical layer</td>
<td>Bits</td>
<td>11010110001010101001001001001101010010</td>
</tr>
</tbody>
</table>
Transport layer protocols are end-to-end protocols only implemented at end systems (aka hosts).
The transport layer

- Multiple application processes are running on a host
- How to deliver data to a given application process?
  - IP provides host-to-host packets delivery but how to deliver packets to a specific application process?
- Each IP packet header has:
  - Source and destination IP addresses
  - Protocol field which species the higher-layer protocol (e.g., UDP = 17, TCP = 6)
- IP demultiplexes data from incoming packets between the transport layer protocols (UDP and TCP) based on the Protocol field value
The transport layer

• The transport layer protocols add a mechanism for the application process identification (port numbers)

• The combination of the following values uniquely identifies a flow in the Internet:
  • Source and destination IP addresses
  • Source and destination port numbers
  • Protocol field value

• E.g., a TCP connection from a TUT computer to www.tut.fi: 130.230.52.139:1080, 130.230.137.61:80, TCP

• Demultiplexing -> delivering incoming data to certain higher-layer entities

• Multiplexing -> gathering data from multiple higher-layer entities, enveloping data with headers (later used for demultiplexing), and passing them to the lower layer
The transport layer

- **Application layer**
  - Messages
  - Byte streams

- **Transport layer**
  - UDP
    - UDP datagrams
  - TCP
    - TCP segments

- **Internet layer**
  - IP

**Demultiplex** based on port numbers in UDP/TCP headers

**Demultiplex** based on the Protocol field value in IP headers
Transmission Control Protocol (TCP)

- Byte-stream-oriented
- Connection-oriented
- Stateful
- Reliable
- Full-duplex
TCP (cont'd)

- Functions:
  - Multiplexing/demultiplexing
  - Ordered data transfer and data segmentation
  - Error control (mandatory)
  - Flow control
  - Congestion control
  - Error control
  - Flow and congestion control
  - Feedback-based

Thus, TCP adds a lot to IP
User Datagram Protocol (UDP)

Message-oriented

Connectionless
  • Establishing a connection before sending data not required
  • Each datagram is handled independently

Stateless
  • Neither the sender nor the receiver has an obligation to keep track of the state of the communication session

Unreliable
  • Data may be lost or delivered out-of-order
  • No ACKs and no retransmissions
UDP (cont'd)

• Functions:
  • Multiplexing/demultiplexing
  • Error control (optional)
• Datagram integrity verification
  • IP computes checksum only for the IP header
  • UDP checksum applies to the entire UDP datagram
• No flow or congestion control
  • UDP can send data as fast as desired, no delivery guarantee
• No feedback messages
  • UDP can be used for both unicast or multicast

Thus, UDP adds little to IP
### UDP vs. TCP

<table>
<thead>
<tr>
<th>UDP</th>
<th>TCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message-oriented</td>
<td>Byte-stream-oriented</td>
</tr>
<tr>
<td>Connectionless</td>
<td>Connection-oriented</td>
</tr>
<tr>
<td>Stateless</td>
<td>Stateful</td>
</tr>
<tr>
<td>Unreliable</td>
<td>Reliable</td>
</tr>
<tr>
<td>Unicast and multicast</td>
<td>Unicast only</td>
</tr>
<tr>
<td>Used by a few user applications</td>
<td>Used by many user applications</td>
</tr>
<tr>
<td>(VoIP, multimedia streaming, etc.)</td>
<td>(WWW, email, FTP, Telnet, etc.)</td>
</tr>
<tr>
<td>Used by many network services</td>
<td>Used by a few network services</td>
</tr>
<tr>
<td>(RIP, SNMP, DNS, etc.)</td>
<td>(e.g., DNS zone transfers)</td>
</tr>
</tbody>
</table>
UDP vs. TCP

Why is there UDP?
- No connection establishment --> no signaling overhead
- No connection state at the end hosts --> few resources are required
- Small header --> small control overhead
- Error control is optional --> suitable for loss-tolerant applications
- No flow or congestion control --> unbounded sending rate
- Simple implementation --> flexibility and scalability

All of those make it a perfect choice for multimedia, delay-sensitive data delivery
APPLICATION LAYER (RTP & RTCP)
RTP: Real-time Transport Protocol

Even though it says „transport protocol”, in fact RTP in an application-layer protocol

• Built on top of the UDP and TCP transport protocols
• Defines a standardized packet format for delivering audio and video content over the Internet
  • Published in 1996 as RFC 1889, obsoleted in July 2003 by RFC 3550

RTP consists of data part and control part
• The latter one called RTCP (RTP Control Protocol)
RTP cont’d

- RTP - thin protocol supporting real-time applications e.g. continuous media (e.g. audio and video)
  - Timing reconstruction, loss detection, security and content identification
- RTCP provides further support for real-time transmission
  - Quality-of-service feedback from receivers to the multicast group
  - Support for syncing of different media streams
RTP cont’d

No standard TCP or UDP port
• UDP transmission is done via an even port from range 16384 – 32767
• The next higher odd port is used for RTCP
  • Dynamic port range makes it difficult for RTP to traverse firewalls

RTP can carry any kind of real-time data
• E.g. interactive audio or video
• Call setup and tear-down usually performed by the SIP protocol
RTP cont’d

Designed as a multicast protocol
- In practice it is mostly used in unicast applications

General scenarios
- One-to-one
- One-to-many
- Many-to-many
- Local transmission (access within one machine)

RTCP exchanges Sender and Receiver Reports
Services provided by RTP:
• Payload-type identification
  • Indicates what kind of content is being carried
• Sequence numbering
  • PDU sequence number
• Time stamping
  • Used for synchronization of the content presentation time
• Delivery monitoring
RTP cont’d

RTP does not provide
• Mechanisms ensuring timely delivery
• Quality of Service (QoS) guarantees

These mechanisms have to be provided by some other entity
RTP cont’d

- Out of order delivery is possible
  - But data necessary for putting the received packets in the correct order is delivered
- No support for flow and congestion control
- RTCP provides information about reception quality
  - Can be used to make local adjustments
  - In case of congestion, the application can decide to lower the data rate
RTP packet

- Consists of and RTP header, optional payload headers and the payload itself
- RTP overhead = 12 bytes
- IP+UDP+RTP overhead = 20+8+12 = 40 bytes
- Advised to keep coded slice sizes close to, but never bigger than the MTU size
  - MTU – largest size of a packet that can be transmitted without being split/recombined, optimizes the payload/header rate
  - Minimize the loss probability of a coded slice due to the loss of a single fragment
  - MTU size usually ~1500 bytes for wired IP links (corresponds to Ethernet MTU)
DVD quality video transmission, 30 frames/s, 720x480 resolution, 3 bytes per pixel

- 31,104,000 bytes/s raw data rate
- 622,080 bytes/s compressed data rate (50x compression)

**MTU = 1500 bytes**
- Required throughput including 40 bytes of overhead:
  - $\frac{622,080}{1460} = 426$ packets/s $\rightarrow$ 640 KB/s

**MTU = 100 bytes**
- Required throughput including overhead:
  - $\frac{622,080}{60} = 10,368$ packets/s $\rightarrow$ 1,036 KB/s

**MTU = 660 bytes $\rightarrow$ ??**
RTP packet – header format

The RTP packet header has the following format:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X| CC | M | PT | sequence number |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         timestamp                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          synchronization source (SSRC) identifier          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            contributing source (CSRC) identifiers                            |
|                                                                                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Based on RFC3550
http://www.ietf.org/rfc/rfc3550.txt
RTP packet – fields

- V - **Version** of RTP, current release uses 2
- P - **Padding** after the payload
- X - presence of header **Ext**ension
- **CSRC count** CC, number of source identifiers
- M – **Marker** bit, e.g. indicate the frame boundaries
- PT – **Payload Type**, media codec of the payload
- **Sequence number**, used for packet-loss detection
- **Timestamp** – used for stream synchronization
- **Source identifiers** (contributing and synchronizing)
Real-Time Transport Protocol

[Stream setup by SDP (frame 1047)]

10. ..... = Version: RFC 1889 Version (2)
...0. ..... = Padding: False
...0. ..... = Extension: False
....0000 = Contributing source identifiers count: 0
0. ..... = Marker: False
Payload type: telephone-event (101)
Sequence number: 6358
[Extended sequence number: 71894]
Timestamp: 4280803975
Synchronization Source identifier: 0x21e7054c (568788300)

RFC 2833 RTP Event

Event ID: DTMF Five 5 (5)
0. ..... = End of Event: False
.0. ..... = Reserved: False
..00 0110 = Volume: 6
Event Duration: 400
RTCP

- RTCP: periodic transmission of control data to all participants
- Multiplexing of data and control packets (separate port numbers) with UDP
- Fraction of the bandwidth allocated to the RTCP is abt. 5%
- Primary function of RTCP is feedback on data distribution quality
RTCP packets

• SR – Sender Report: Transmission and reception statistics from active senders
• RR – Receiver Report: Reception statistics from participants that are not active senders
• SDES - Source description items, e.g. CNAME (Canonical Name – RTP source identifier)
• BYE - Indicates end of participation
• APP - Application specific functions
RTCP – feedback mechanism

Sender and Receiver Reports (SR & RR)
Timestamps allowing to calculate the RTT
• RTT = T4-T3+T2-T1
Packet counts
Inter-arrival jitter (variation in delay)
Fraction of packets lost, cumulative number of packet lost
Number of packets expected to have been received
Available bandwidth estimation
RTCP – feedback mechanism

Feedback useful for the sender, other receivers and third-party monitors

- The sender may modify its transmissions parameters
- Receivers determine whether problems are local, regional or global
- Network managers use monitors receiving only the RTCP packets for performance evaluation

Cumulative counts in both the sender information and receiver report

- Calculate differences between any two reports
- Measurements over both short and long time periods
- Provide resilience against the loss of a report
RTCP – available statistics

Using the SR and RR the following info can be obtained:
• Packet loss rate over interval between two RR
• Number of packets expected during interval
• Packet loss fraction over interval – ratio of the two above
• Loss rate per second
• Number of packets received -> number of packets expected minus the number lost
• Statistical validity of any loss estimates
  • 1 out of 5 packets lost less significant than 200 out of 1000
• Apparent throughput available to one receiver
• Interarrival jitter – a short-term measure of network congestion
CROSS-LAYERING
Cross-layering

- The TCP/IP is based on a layered model
- Layered model defines a restrictive interaction between layers
  - Every layer is allowed to interact only with adjacent layers
  - Communication resulting from *overjumping* is prohibited
- The TCP/IP protocol suite was developed primary for wired networks
- TCP/IP protocols perform *suboptimal or poorly* over *wireless networks*
- Wireless networks have pushed the activities around cross-layering
Cross-layering

- Cross-layering approach neglects or weakens this protocol design rule for the purpose of performance optimization, resource preservation, or error/delay tolerance.
- The basic idea behind the cross-layering is to optimize TCP/IP performance over wireless networks (in terms of data rate, losses, delay, jitter, battery power, etc.).
- The background for considering cross-layered approaches is derived from interlayer dependencies.
- Currently mostly a research topic, no commercial implementations available yet.
Cross-layering

New interfaces

Direct communication between non-adjacent layers

Cross-layer performance control system

Sharing data across layers and cross-layer performance optimization

Shared database & control system

No more layering

Completely new abstractions and new interfaces
Thank you

Q&A