

COMPUTER NETWORKS, MULTIMEDIA

013

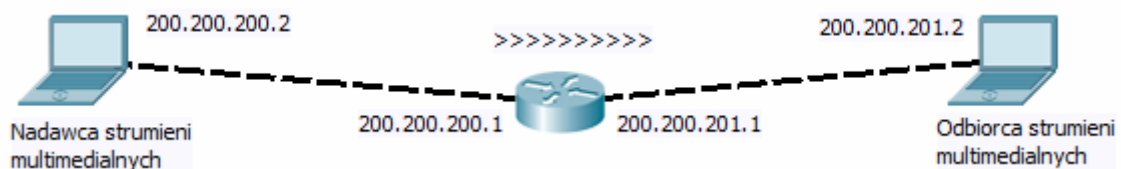
Subject:

Multimedia network streaming:

- RTP (Real Time Protocol) - IP multicast traffic
- RTP unicast IP traffic (for MPEG Transport Stream and other muxes)
- RTSP (Real Time Streaming Protocol)
- Legacy UDP streaming
- Multimedia streams over HTTP
- MMS (Microsoft Media Server)

Task A: RTP (Real Time Protocol) IP, multicast traffic.

1. RTP protocol packets are usually sent over the UDP - providing core functionality for defining video frames, synchronization, media streaming (audio + video) etc. However, that knowledge does not provide specifics on the stream itself (codec variants, encoded signal parameters, etc.). To restore signal on destination host, it is necessary to add such information. Usually it is done by adding to the RTP protocol support a SDP (Session Description Protocol)
2. Prepare installation consisting of two Cisco router and PC computer:



Check a status of firewall software on PC computers, making sure that it will not interfere with the experiments.

3. Configure routers to support both unicast and multicast IP routing:


```
Router (config) #ip routing
Router (config) #ip multicast-routing
Router (config) # int fa 0 / 0
Router (config-if) # ip address 200.200.200.1 255.255.255.0
Router (config-if) #ip pim dense-mode
Router (config-if) #no shut
Router (config) # int fa 0 / 1
Router (config-if) # ip address 200.200.201.1 255.255.255.0
Router (config-if) #ip pim dense-mode
Router (config-if) #no shut
```
4. All tasks will be performed using tools VideoLAN (VLC Player) used as receivers and transmitters of multimedia streams.

5. IP multicast streaming:
On the first PC run IP multicast source.
VLC: use the following command:

```
vlc.exe "c:\File" :sout={rtp#dst=224.1.1.1,port=6666,mux=ts,ttl=10} :sout-all: :sout-keep --repeat
```

where file is the path to the file with the material for streaming and mux is a kind of libav format. Ts is selected MPEG Transport Stream - a variant designed for MPEG streaming (including the extension of the Session Description Protocol). It is necessary to emit an IP multicast datagrams with TTL greater than a the number of routers planned to go through (to reach the destination).

6. On the second turn on the PC turn a Multicast RTP Multicast RTP media player on:

```
vlc.exe: rtp://224.1.1.1: 6666
```

It is possible to specify additional IP unicast source to exclude other sources of some multicast group (formally it's: Source-Specified Multicast, SSM):

```
vlc.exe rtp://200.200.200.2@224.1.1.1:6666
```

7. Run the Wireshark on a PC computer. Observe a traffic with multicast IP / UDP transmission.

Task B: RTP unicast IP traffic (MPEG Transport Stream)

1. In opposition to previous case (IP multicast), IP unicast UDP/RTP transmission requires the sending of IP packets as the recipient stream.
2. Start the transmission with following command line:

```
vlc.exe " c:\file" :sout={rtp#dst=200.200.201.2,port=6666,mux=ts;ttl=10} :sout-all :sout-keep
```

UDP datagrams will now be forwarded to the recipient host, and there either be lost or received by some application (if it exists and waiting on UDP socket). The stream is marked with destination UDP port number, so the recipient must specify it when connecting to proper socket:

```
vlc.exe rtp://200.200.201.2:6666
```

3. Observe (Wireshark) the traffic associated with the IP unicast / UDP transmission. Check whether it is possible to connect two client applications to the same stream. Start at the same time a similar transmission using a different UDP port.

4. Set the transmission again, but using personally selected streams (supplied under RTP by name):

Broadcast:

```
vlc.exe "c:file"  
:sout=#rtp{dst=200.200.201.2,name=streamname,port=6666,mux=ts,ttl=10}  
:sout-all :sout-keep
```

Stream receiving command:

```
vlc.exe rtp://200.200.201.2:6666/streamname
```

Task C: RTSP (Real Time Streaming Protocol)

1. The use of RTSP does not require a use of mutex. The default UDP port number received by the customer (recipient) for RTSP is 554.

Broadcast:

```
vlc.exe "c:\file"  
:sout=#rtp{dst=200.200.201.2,ttl=10,sdp=rtsp://200.200.200.2:554/}  
:sout-all :sout-keep
```

where 200.200.200.2 is the sender's IP address (SDP is transmitted by the sender host in RTSP).

Stream receiving command:

```
vlc.exe rtsp://200.200.200.2:554/
```

Note: Playback requires picking a stream which has already reached the destination computer .

Note: The character '/' at the end of the command is required and means a stream under undefined name. If the recipient located on the same host as the broadcaster does not specify the port number, port 554 is used.

2. With names of streams additionally used, the command will look as follows:

```
vlc.exe "c:\file"  
:sout=#rtp{dst=200.200.201.2,ttl=10,sdp=rtsp://200.200.200.2:554/stream}  
:sout-all :sout-keep
```

Stream receiving command:

```
vlc.exe rtsp://200.200.200.2:554/stream
```

3. Start the transmission and watch the traffic associated with it (Wireshark). Again - check whether it is possible to connect two client applications to the same flow.

Task D: MPEG-TS streaming directly over UDP (without RTP)

1. Complete abandonment of RTP (UDP legacy mode) requires an appropriate mux in a stream (to meet requirements of MPEG Transport Stream).

```
vlc.exe "c:\file"  
:sout=#udp{dst=200.200.201.2:6666,mux=ts,ttl=10} :sout-all :sout-keep
```

The syntax used in the URL to receive a legacy UDP stream:

```
udp://ip_source:port_source@ip_dest:port_dest
```

You can skip *ip_dest* (UDP datagrams have already been delivered to the recipient's destination computer):

```
vlc.exe udp://200.200.200.2@200.200.201.2:6666  
vlc.exe udp://200.200.200.2@:6666
```

or skip the source specification:

```
vlc.exe udp://@200.200.201.2:6666
```

2. Start the transmission. As before, check whether it is possible to connect two client applications to the same flow.

Task E: HTTP streaming transmissions in the client-server architecture

1. A use of client-server architecture requires to run a server (sender) first, waiting for client connections. The built-in HTTP server package VideoLAN lets you use the HTTP service for streaming. Again, activities related to streaming control (muxer) will be set over MPEG-TS.

Broadcast:

```
vlc.exe "c:\file" :sout=#http{mux=ts,dst=:6666} :sout-all :sout-keep
```

where 6666 this time the TCP port for the HTTP service (data streaming on request), '/' represents the resource's ID within the HTTP URL.

Note: an empty field before ":" indicates a localhost.

Reception: vlc.exe:

`http://200.200.200.2:6666/`

where 200.200.200.2 is an IP address of a SENDER.

2. Start the transmission and watch (Wireshark) the traffic associated with it. Check whether it is possible to connect two client applications to the same flow.

Task F: Microsoft Media Server live network streaming format

1. Streaming compatible with MMS (also in a client-server architecture) requires the stream to be re-encoded to Windows Media Server multimedia stream.

Broadcasting and conversion into the required format in real time:

```
vlc.exe "c:\file"  
:sout=#transcode{vcodec=DIV3,vb=800,acodec=mp3,ab=128,channels=2,  
samplerate=44100}:std{access=mmsh,mux=asfh,dst=0.0.0.0:6666}  
:sout-all :sout-keep
```

where 6666 is the port number the MMC service, 0.0.0.0 allows hosts to receive a stream.

Reception by VideoLAN (requirement is to use the MMS-Helper which is an extension of the MMC unifying VideoLan):

```
vlc.exe mmsh://200.200.200.2:6666
```

where 200.200.200.2 address is a SENDER IP address.

2. Start Windows Media Player application on a Windows PC computer. In the "File" menu select "Open URL" and enter the following address:

```
mms://200.200.200.2:6666
```

3. Watch (Wireshark) a transmission traffic. Check whether it is possible to connect two client applications to the same flow.

Task G: VideoLAN – other streaming operations

Run stream (configured sender + receiver) in the following situations:

1. DUPLICATE: two windows on-screen, window + stream via UDP:

```
vlc.exe "c:\file" :sout=#duplicate{  
dst=display,
```

```
dst=display
}
:sout-all :sout-keep
```

```
vlc.exe "c:\file" :sout=#duplicate{
dst=rtp{dst=200.200.201.2,port=6666,mux=ts,ttl=10},
dst=display
}
:sout-all :sout-keep
```

2. DUPLICATE: two streams at the same network protocol:

```
vlc.exe "c:\file" :sout=#duplicate{
dst=rtp{dst=200.200.201.2,port=6666,mux=ts,ttl=10},
dst=rtp{dst=200.200.201.2,port=7777,mux=ts,ttl=10}
}
:sout-all :sout-keep
```

3. DUPLICATE two streams in the same network and transport protocol but use two transmission modes (IP unicast and IP multicast):

```
vlc.exe "c:\file" :sout=#duplicate{
dst=rtp{dst=200.200.201.2,port=6666,mux=ts,ttl=10},
dst=rtp{dst=224.1.1.1,port=7777,mux=ts,ttl=10}
}
:sout-all :sout-keep
```

4. DUPLICATE: two streams in a variety of network protocols:

```
vlc.exe "c:\file" :sout=#duplicate{
dst=rtp{dst=200.200.201.2,port=6666,mux=ts,ttl=10},
dst=http{dst=:6666/,mux=ts}
}
:sout-all :sout-keep
```

Run the above examples one by one, each time choosing the proper configuration of recipients (like in the previous task).

5. Playback YouTube resources redirecting it into a network stream:

```
vlc.exe https://www.youtube.com/watch?v=BEG-ly9tQGk
:sout=#duplicate{dst=display,dst=rtp{dst=200.200.201.2,port=6666,mux=ts,ttl=10}
} :sout-all :sout-keep
```

6. Streaming the video from USB camera – and transcoding it into MPG stream (connect the camera running the command):

```
vlc dshow:// :dshow-size="320x240"  
:sout=#transcode{vcodec=mpgv,vb=1024,scale=1,acodec=mp3,ab=192,channels  
=2}:rtp{dst=224.1.1.1,port=6666,mux=ts,tfl=10} :sout-keep
```

7. A screen capture:

```
vlc screen://
```

```
vlc screen:// :screen-width=640 :screen-height=480 :screen-follow-mouse
```

8. Capturing screen and streaming it to a file with mp4 + mp2/mp3 encoding:

```
vlc.exe screen:// :screen-fps=24 :screen-width=640:screen-height=480 :screen-  
follow-mouse :dshow-caching=150  
:sout=#transcode{vcodec=h264,soverlay,acodec=mp2a,ab=128,scale=1,channels  
=2,deinterlace,audio-sync}:standard{access=file,mux=ts,dst="c:\out.mp4"}
```