

Streaming de áudio e vídeo a pedido e em tempo real

Filipa Rebelo, Luis Araújo e Luís Fernandes

Universidade do Minho, Departamento de Informática, 4710-057 Braga, Portugal

email: {pg53624,pg54004,pg54022}@alunos.uminho.pt

1 Etapa 1. Streaming HTTP simples sem adaptação dinâmica de débito

1.1 Questão 1

Capture três pequenas amostras de tráfego no link de saída do servidor, respetivamente com 1 cliente (VLC), com 2 clientes (VLC e Firefox) e com 3 clientes (VLC, Firefox e ffplay). Identifique a taxa em bps necessária (usando o `ffmpeg -i videoA.mp4` e/ou o próprio `wireshark`), o encapsulamento usado e o número total de fluxos gerados. Comente a escalabilidade da solução. Ilustre com evidências da realização prática do exercício (ex: capturas de ecrã)

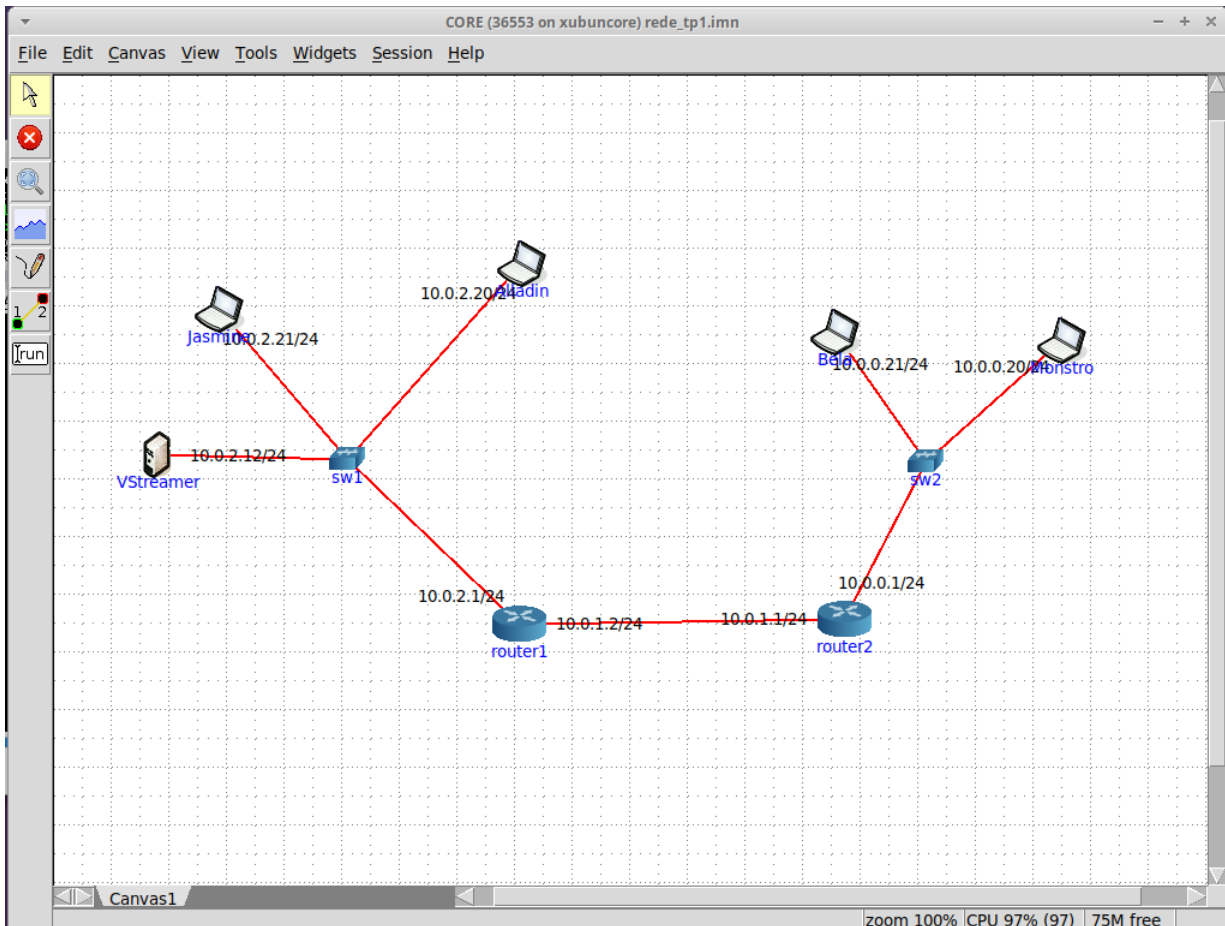


Fig. 1. Topologia Core

Através da análise da figura 2, usando o ffmpeg, é possível verificar que o bitrate para a transmissão do vídeo videoA.mp4 é de 18 kb/s.

```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'videoA.mp4':
Metadata:
  major_brand      : isom
  minor_version    : 512
  compatible_brands: isomiso2avc1mp41
  encoder          : Lavf58.29.100
Duration: 00:00:05.90, start: 0.000000, bitrate: 21 kb/s
Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 200x150, 18 kb/s, 20 fps, 20 tbr, 10240 tbn, 40 tbc (default)
Metadata:
  handler_name     : VideoHandler
```

Fig. 2. ffmpeg -i videoA.mp4

É possível verificar o encapsulamento usado através da figura 3. Foi usado o protocolo Ethernet na camada de ligação, IP na camada de rede, TCP na camada de transporte e HTTP na camada aplicacional.

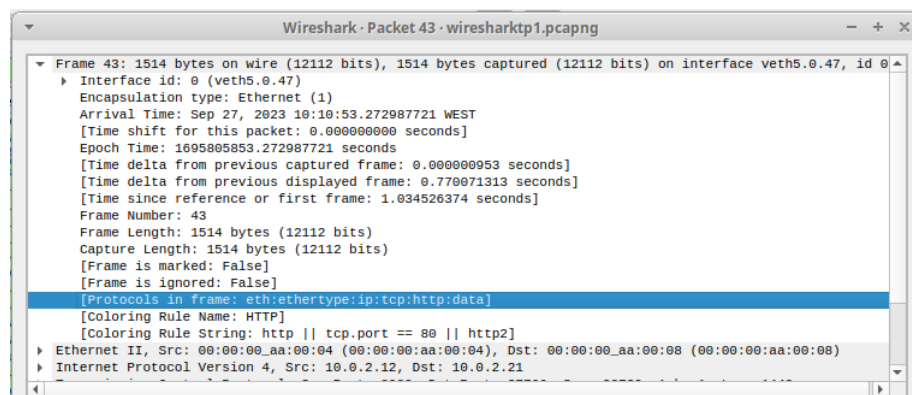


Fig. 3. Encapsulamento

Wireshark - Protocol Hierarchy Statistics - tp1_1.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	686	100.0	450664	135 k	0	0	0
Ethernet	100.0	686	2.1	9604	2884	0	0	0
Internet Protocol Version 6	0.4	3	0.0	120	36	0	0	0
Open Shortest Path First	0.4	3	0.0	108	32	3	108	32
Internet Protocol Version 4	99.3	681	3.0	13620	4090	0	0	0
Transmission Control Protocol	97.4	668	94.7	426584	128 k	622	368034	110 k
Hypertext Transfer Protocol	6.7	46	12.8	57648	17 k	46	57648	17 k
Open Shortest Path First	1.9	13	0.1	572	171	13	572	171
Address Resolution Protocol	0.3	2	0.0	56	16	2	56	16

Fig. 4. 1 cliente-VLC

Wireshark - Protocol Hierarchy Statistics - tp1_2.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	291	100.0	195718	275 k	0	0	0
Ethernet	100.0	291	2.1	4074	5728	0	0	0
Internet Protocol Version 4	100.0	291	3.0	5820	8184	0	0	0
Transmission Control Protocol	99.0	288	94.9	185692	261 k	268	159520	224 k
Hypertext Transfer Protocol	6.9	20	13.1	25722	36 k	20	25722	36 k
Open Shortest Path First	1.0	3	0.1	132	185	3	132	185

Fig. 5. 2 clientes-VLC e Firefox

Wireshark - Protocol Hierarchy Statistics - tp1_3.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	931	100.0	617982	393 k	0	0	0
Ethernet	100.0	931	2.1	13034	8299	0	0	0
Internet Protocol Version 6	0.1	1	0.0	40	25	0	0	0
Open Shortest Path First	0.1	1	0.0	36	22	1	36	22
Internet Protocol Version 4	99.9	930	3.0	18600	11 k	0	0	0
Transmission Control Protocol	99.2	924	94.8	586008	373 k	855	499173	317 k
Hypertext Transfer Protocol	7.4	69	13.8	85341	54 k	69	85341	54 k
Open Shortest Path First	0.6	6	0.0	264	168	6	264	168

Fig. 6. 3 clientes-VLC, Firefox e ffmpeg

Ao usar o *ffmpeg*, identificamos que, embora a taxa de bits do vídeo seja de aproximadamente 18kb/s, conforme ilustrado na Figura 2, na Figura 4 revela um overhead TCP de cerca de 128kb/s, mesmo com apenas um fluxo gerado, para transportar o tráfego HTTP do vídeo.

Quanto à taxa de transmissão, ao enviar o vídeo A para um único cliente, a taxa foi de 17kb/s. Com dois clientes, esta incrementou para 36kb/s, e com três, para 54kb/s. Isso indica que a taxa de transmissão e o overhead do TCP crescem proporcionalmente ao número de clientes. Para um, dois e três fluxos gerados, o tráfego do encapsulamento TCP foi de 128kb/s, 261kb/s e 373kb/s, respetivamente.

Através da figura 7 é possível visualizar a criação de 3 fluxos gerados.

Wireshark · Conversations · tp1_3.pcapng

Ethernet · 4		IPv4 · 4		IPv6 · 1		TCP · 3		UDP							
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A		
10.0.2.12	8080	10.0.2.21	48108	308	205 k	154	195 k	154	10 k	0.000000	12.5626	124 k	6472		
10.0.2.12	8080	10.0.0.21	48294	308	205 k	154	195 k	154	10 k	0.000104	12.5627	124 k	6472		
10.0.2.12	8080	10.0.0.20	47972	308	205 k	154	195 k	154	10 k	0.000198	12.5632	124 k	6472		

Fig. 7. Fluxos TCP com 3 clientes

Por fim, quanto à escalabilidade é possível concluir que a solução apresentada não seja escalável, uma vez que existem tantos fluxos gerados e com a mesma quantidade de dados transmitidos, quantos clientes adicionados à transmissão do vídeo.

2 Etapa 2.Streaming adaptativo sobre HTTP (MPEG-DASH)

2.1 Questão 2

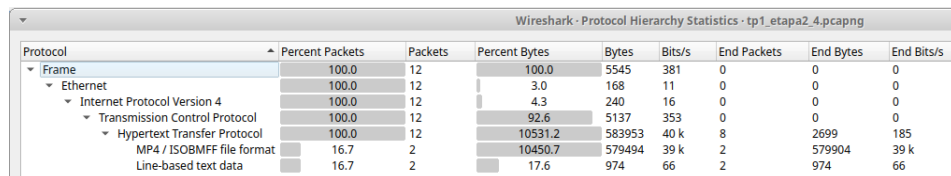
Diga qual a largura de banda necessária, em bits por segundo, para que o cliente de streaming consiga receber o vídeo no firefox e qual a pilha protocolar usada neste cenário.

```

<!--
MPD file Generated with GPAC version 0.5.2-DEV-revVersion: 0.5.2-426-gc5ad4e4dfsg5-5 at 2023-10-01T17:43:28.760Z
-->
<MPD minBufferTime="PT1.500S" type="static" mediaPresentationDuration="PT0H0M7.900S" maxSegmentDuration="PT0H0M0.500S" profiles="urn:mpeg:dash:profile:full:2011">
  <ProgramInformation moreInformationURL="http://gpac.sourceforge.net">
    <Title>video_manifest.mpd generated by GPAC</Title>
  </ProgramInformation>
  <Period duration="PT0H0M7.900S">
    <AdaptationSet segmentAlignment="true" bitstreamSwitching="true" maxWidth="640" maxHeight="480" maxFrameRate="30" par="4:3" lang="und">
      <SegmentList>
        <Initialization sourceURL="video_manifest_init.mpd"/>
      </SegmentList>
      <Representation id="1" mimeType="video/mp4" codecs="avc3.64000c" width="200" height="150" frameRate="30" sar="1:1" startWithSAP="0" bandwidth="104058"></Representation>
      <Representation id="2" mimeType="video/mp4" codecs="avc3.64001e" width="480" height="360" frameRate="30" sar="1:1" startWithSAP="0" bandwidth="286995"></Representation>
      <Representation id="3" mimeType="video/mp4" codecs="avc3.64001e" width="640" height="480" frameRate="30" sar="1:1" startWithSAP="0" bandwidth="482770"></Representation>
    </AdaptationSet>
  </Period>
</MPD>

```

Fig. 8. Largura de banda necessária para transmitir os 3 vídeos



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	12	100.0	5545	381	0	0	0
Ethernet	100.0	12	3.0	168	11	0	0	0
Internet Protocol Version 4	100.0	12	4.3	240	16	0	0	0
Transmission Control Protocol	100.0	12	92.6	5137	353	0	0	0
Hypertext Transfer Protocol	100.0	12	10531.2	583953	40 k	8	2699	185
MP4 / ISOBMFF file format	16.7	2	10450.7	579494	39 k	2	579904	39 k
Line-based text data	16.7	2	17.6	974	66	2	974	66

Fig. 9. Pilha protocolar usada

Pela figura 8 é possível observar que a largura de banda mínima necessária para que o cliente de streaming receba o vídeo é de 104 kb/s (104058 bps). A largura de banda necessária para o vídeo com melhor qualidade é de 482 kb/s (482770 bps). A pilha protocolar pode ser visível na figura 9, [Ethernet[IP[TCP[HTTP]]]]].

2.2 Questão 3

Ajuste o débito dos links da topologia de modo que o cliente no portátil Bela exiba o vídeo de menor resolução e o cliente no portátil Alladin exiba o vídeo com mais resolução. Mostre evidências.

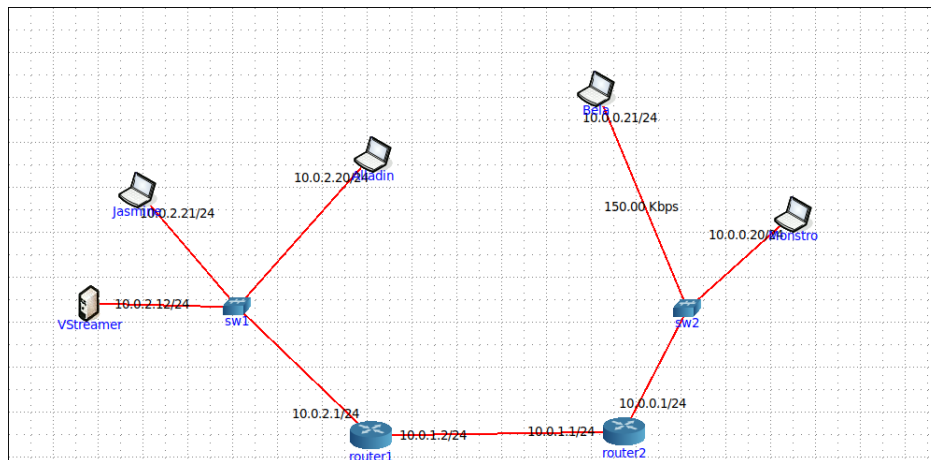


Fig. 10. Topologia com a largura de banda limitada no PC da Bela

http					
No.	Time	Source	Destination	Protocol	Length Info
26	15.849153915	10.0.0.21	10.0.2.12	HTTP	381 GET /favicon.ico HTTP/1.1
28	15.849259481	10.0.0.21	10.0.0.21	HTTP	741 HTTP/1.1 404 Not Found (text/html)
36	16.099931390	10.0.0.21	10.0.2.12	HTTP	411 GET /videoB_640_480_1000k_dash.mp4 HTTP/1.1
120	18.045664783	10.0.0.21	10.0.2.12	HTTP	410 GET /videoB_200_150_200k_dash.mp4 HTTP/1.1
462	28.592640506	10.0.2.12	10.0.0.21	MP4	221
790	43.677851232	10.0.2.12	10.0.2.12	HTTP	411 GET /videoB_640_480_1000k_dash.mp4 HTTP/1.1
1305	63.965511235	10.0.0.21	10.0.2.12	HTTP	411 GET /videoB_640_480_1000k_dash.mp4 HTTP/1.1
1842	85.306813205	10.0.0.21	10.0.2.12	HTTP	411 GET /videoB_640_480_1000k_dash.mp4 HTTP/1.1
2376	131.868671377	10.0.2.20	10.0.2.12	HTTP	381 GET /favicon.ico HTTP/1.1
2378	131.868759846	10.0.2.12	10.0.2.20	HTTP	741 HTTP/1.1 404 Not Found (text/html)
2386	132.164690809	10.0.2.20	10.0.2.12	HTTP	411 GET /videoB_640_480_1000k_dash.mp4 HTTP/1.1
2741	132.166088541	10.0.2.12	10.0.2.20	MP4	615

Fig. 11. Pedidos GET e transmissão do vídeo para os PCs da Bela(azul) e do Aladin(amarelo)

O PC da Bela foi limitado com uma largura de banda de 150 kb/s(figura 10). Deste modo a partir da figura 11 é possível observar que a Bela, primeiro faz o pedido GET da versão de maior qualidade mas depois devido à sua largura de banda, restringida, faz o pedido GET da versão de menor qualidade e reproduz o mesmo. Como a largura de banda do PC do Alladin não está restringida no CORE, o vídeo que este recebe, é o de maior qualidade.

2.3 Questão 4

Descreva o funcionamento do DASH neste caso concreto, referindo o papel do ficheiro MPD criado

O DASH (Dynamic Adaptive Streaming over HTTP) é uma técnica de streaming que permite a entrega de conteúdo multimédia através da web, adaptando-se dinamicamente às condições da rede e às capacidades do dispositivo de reprodução. O DASH funciona dividindo o conteúdo multimédia numa série de pequenos segmentos, cada um com uma duração específica, permitindo que o cliente selecione o segmento apropriado com base na sua largura de banda disponível.

O ficheiro MPD (Media Presentation Description) é um componente central do DASH. Este fornece meta dados sobre o conteúdo multimédia, incluindo informações sobre os segmentos disponíveis, resoluções, taxas de bits e outros parâmetros. O cliente DASH usa o MPD para tomar decisões sobre quais segmentos transferir e em que ordem o deve fazer.

Em resumo, este MPD(video_manifest.mpd) apresenta três versões diferentes do vídeo, visíveis na figura 8, cada uma com diferentes resoluções e largura de banda. O cliente DASH usa este MPD para decidir qual versão transferir com base nas condições da rede e nas capacidades do dispositivo.

3 Etapa 3. Streaming RTP/RTCP unicast sobre UDP e multicast com anúncios SAP

3.1 Questão 5

Compare o cenário unicast aplicado com o cenário multicast. Mostre vantagens e desvantagens na solução multicast ao nível da rede, no que diz respeito a escalabilidade (aumento do nº de clientes) e tráfego na rede. Tire as suas conclusões.

Wireshark - Protocol Hierarchy Statistics - tp1_etapa3_unicast.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	2742	100.0	2165285	182 k	0	0	0
Ethernet	100.0	2742	1.8	38388	3236	0	0	0
Internet Protocol Version 6	0.6	17	0.0	680	57	0	0	0
User Datagram Protocol	0.1	2	0.0	16	1	0	0	0
Multicast Domain Name System	0.1	2	0.0	282	23	2	282	23
Open Shortest Path First	0.4	10	0.0	360	30	10	360	30
Internet Control Message Protocol v6	0.2	5	0.0	80	6	5	80	6
Internet Protocol Version 4	99.1	2717	2.5	54340	4580	0	0	0
User Datagram Protocol	94.5	2592	1.0	20736	1748	0	0	0
Data	94.3	2587	93.1	2016836	170 k	2587	2016836	170 k
ADwin configuration protocol	0.2	5	0.0	260	21	5	260	21
Open Shortest Path First	1.7	47	0.1	2068	174	47	2068	174
Internet Control Message Protocol	2.8	78	1.4	31015	2614	78	31015	2614
Address Resolution Protocol	0.3	8	0.0	224	18	8	224	18

Fig. 12. Hierarquia Protocolos captura Unicast

Wireshark - Protocol Hierarchy Statistics - tp1_etapa3_multicast.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	3063	100.0	2465995	179 k	0	0	0
Ethernet	100.0	3063	1.7	42882	3118	0	0	0
Internet Protocol Version 6	0.6	18	0.0	720	52	0	0	0
Internet Protocol Version 4	99.4	3045	2.5	60964	4433	0	0	0
User Datagram Protocol	98.9	3029	1.0	24232	1762	0	0	0
Session Announcement Protocol	0.7	21	0.3	6804	494	0	0	0
Session Description Protocol	0.7	21	0.3	6300	458	21	6300	458
Real-Time Transport Protocol	93.2	2855	90.2	2225020	161 k	0	0	0
MP4V-ES	93.2	2855	88.8	2190760	159 k	2855	2190760	159 k
Real-time Transport Control Protocol	0.7	21	0.0	588	42	21	588	42
Data	4.3	132	4.2	103709	7542	132	103709	7542
Internet Group Management Protocol	0.5	16	0.0	256	18	16	256	18

Fig. 13. Hierarquia Protocolos captura Multicast

O Unicast oferece uma comunicação direta e simples entre servidor e destinatário. Este permite um controlo eficaz na largura de banda e é amplamente compatível com a maioria das redes e dispositivos. Além disso, proporciona desempenho e latência previsíveis e a capacidade de personalizar o conteúdo para cada utilizador. Porém, se muitos usuários em locais diferentes quiserem aceder a um conteúdo ao mesmo tempo (por exemplo, assistir a um vídeo ao vivo), o servidor terá que enviar uma cópia separada do conteúdo para cada utilizador, o que pode congestionar a rede.

O Multicast é streaming um para muitos. Este envia informações (vídeo, neste caso) de um ponto para vários "endpoints" simultaneamente. É uma espécie de solução única que não se ajusta ao ecrã em que está a ser visualizada, a menos que tenha mais que um fluxo Multicast enviado pela rede. Por outro lado, o modelo de streaming Multicast não é facilmente escalável para redes heterogéneas na Internet. O Multicast assume que todos os "end systems" têm a mesma capacidade de largura de banda e que usam um dispositivo similar para aceder ao conteúdo. Isso dificulta a escalabilidade e a transmissão aos clientes que têm larguras de banda mais baixas ou flutuantes ou dispositivos diferentes daqueles para os quais o streaming de conteúdo se destina.

Comparando os valores da figura 12 que dizem respeito à captura do wireshark no link de saída do servidor no cenário Unicast, com os da figura 13 que também são de uma captura no link de saída do servidor mas neste caso num cenário Multicast, verificamos que o tráfego em Bits/s é basicamente o mesmo. No cenário Unicast temos apenas um endpoint a receber o vídeo e no Multicast quatro, mas devido ao modo de funcionamento do Multicast em que os pacotes só são duplicados para os vários endpoints no último link em que tal seja possível, até esse link não existe duplicação de pacotes. Evita-se deste modo o congestionamento desnecessário da rede.

4 Conclusão

Durante o trabalho foi possível aplicar conhecimentos obtidos nas aulas teóricas tais como streaming de áudio e vídeo, identificando que soluções não escaláveis têm desempenho reduzido com mais clientes. Ao experimentar o streaming adaptativo (MPEG-DASH), enfrentamos desafios com a largura de banda. Na última etapa, exploramos streaming RTP/RTCP unicast e multicast. Concluímos o trabalho, atingindo as metas, especialmente sobre protocolos de streaming e as diferenças entre soluções Unicast e Multicast.