



UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Tecnologia

RAFAEL FERNANDO DIORIO

ENCAMINHAMENTO, DISTRIBUIÇÃO E PREDIÇÃO MULTIMÍDIA EM  
REDES DEFINIDAS POR SOFTWARE

LIMEIRA  
2017

RAFAEL FERNANDO DIORIO

ENCAMINHAMENTO, DISTRIBUIÇÃO E PREDIÇÃO MULTIMÍDIA EM  
REDES DEFINIDAS POR SOFTWARE

*Tese apresentada à Faculdade de  
Tecnologia da Universidade Estadual de  
Campinas como parte dos requisitos  
exigidos para a obtenção do título de  
Doutor em Tecnologia, na Área de  
Sistemas de Informação e Comunicação.*

*Supervisor/Orientador:* Prof. Dr. Varese Salvador Timóteo

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL  
DA TESE DEFENDIDA PELO ALUNO RAFAEL  
FERNANDO DIORIO E ORIENTADA PELO PROF. DR.  
VARESE SALVADOR TIMÓTEO.

LIMEIRA  
2017

**Agência(s) de fomento e nº(s) de processo(s): CAPES**

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Faculdade de Tecnologia  
Felipe de Souza Bueno - CRB 8/8577

D623e Diorio, Rafael Fernando, 1983-  
Encaminhamento, distribuição e predição multimídia em redes definidas por software / Rafael Fernando Diorio. – Limeira, SP : [s.n.], 2017.

Orientador: Varese Salvador Timoteo.  
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Tecnologia.

1. Comunicações multimídia. 2. Sistemas multimídia. 3. Redes definidas por software (Tecnologia de rede de computador). I. Timoteo, Varese Salvador, 1972-. II. Universidade Estadual de Campinas. Faculdade de Tecnologia. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Delivery, distribution and prediction of multimedia contents in software-defined networks

**Palavras-chave em inglês:**

Multimedia communications

Multimedia systems

Software-defined networking (Computer network technology)

**Área de concentração:** Sistemas de Informação e Comunicação

**Titulação:** Doutor em Tecnologia

**Banca examinadora:**

Varese Salvador Timoteo [Orientador]

Edson Luiz Ursini

Paulo Sérgio Martins Pedro

Kalinka Regina Lucas Jaquie Castelo Branco

Luiz Henrique Bonani do Nascimento

**Data de defesa:** 01-12-2017

**Programa de Pós-Graduação:** Tecnologia

## FOLHA DE APROVAÇÃO

Abaixo se apresentam os membros da Comissão Examinadora da Sessão Pública de Defesa de Tese para obtenção do título de Doutor em Tecnologia, na área de Sistemas de Informação e Comunicação, a que se submeteu o aluno Rafael Fernando Diorio, em 01 de dezembro de 2017 na Faculdade de Tecnologia – FT/UNICAMP, em Limeira/SP.

**Prof. Dr. Varese Salvador Timóteo**

Presidente da Comissão Examinadora

**Prof. Dr. Edson Luiz Ursini**

FT/UNICAMP

**Prof. Dr. Paulo Sérgio Martins Pedro**

FT/UNICAMP

**Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco**

ICMC/USP

**Prof. Dr. Luiz Henrique Bonani do Nascimento**

CECS/UFABC

A Ata da Defesa, com as respectivas assinaturas dos membros da Comissão Examinadora, encontra-se no processo de vida acadêmica do aluno na Universidade.

## DEDICATÓRIA

Dedico este trabalho, em especial, aos meus pais (Elizie e Sérgio) e à minha filha (Giovana).

## AGRADECIMENTOS

Em primeiro lugar, agradeço a Deus por ter “conseguido chegar até aqui”. Sou muito grato por tudo que me foi proporcionado até o “presente momento”.

Não menos importante, agradeço à minha família, minha namorada e meus amigos por suportarem minha “ausência” ao longo deste trabalho. Sou muito grato pelo amor, carinho, paciência e suporte proporcionado por vocês.

Ao meu pai, em memória, registro meu profundo agradecimento pela formação pessoal e profissional que me foi proporcionada. Obrigado por tudo pai!

Agradeço, também, ao meu orientador, Prof. Varese, pelos vários anos (do Mestrado ao Doutorado) de trabalho em conjunto. Sou muito grato pela amizade, confiança e suporte ao longo de nosso trabalho.

Por fim, registro meus agradecimentos ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP), pela possibilidade de me dedicar exclusivamente ao Doutorado (enquanto afastamento para qualificação), à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pela bolsa de estudos concedida durante a realização deste trabalho, e aos colaboradores e profissionais da Faculdade de Tecnologia (FT) da Universidade Estadual de Campinas (UNICAMP), em especial, pelo suporte acadêmico proporcionado ao longo deste trabalho.

## RESUMO

Nos últimos anos, aplicações e serviços multimídia, tais como para o acesso e compartilhamento de conteúdos de áudio e vídeo ou para realização de jogos *online*, dentre outras, tornaram-se cada vez mais populares. Por outro lado, esses novos serviços e aplicações tornaram o fluxo de tráfego de rede, além de mais intenso, mais detalhado e complexo, exigindo abordagens específicas para comunicações multimídia. Diante desse cenário, este trabalho propõe e explora, enquanto recursos de um *gateway* multimídia no âmbito de Redes Definidas por *Software* (SDN) OpenFlow, algumas abordagens que objetivam aprimorar comunicações multimídia no que se refere ao encaminhamento, à distribuição e à predição de seus conteúdos ao longo de rede. Para esse propósito, no âmbito de um *gateway* multimídia em SDN, os seguintes aspectos são especialmente discutidos neste trabalho: (I) identificação, classificação e encaminhamento diferenciado de fluxos de tráfego multimídia; (II) encaminhamento por fluxos com suporte à Qualidade de Serviço (QoS); (III) cache de conteúdos multimídia; e, (IV) predição de conteúdos multimídia. Nesse contexto, para efeitos de experimentação e avaliação das abordagens empregadas neste trabalho, um ambiente de emulação e um ambiente de simulação são utilizados para a obtenção de resultados experimentais. Em linhas gerais, os resultados obtidos mostram que tais abordagens, enquanto recursos de um *gateway* multimídia no âmbito de SDNs OpenFlow, são capazes de fornecer recursos para aprimorar comunicações multimídia ao longo da rede, nesse caso, com ênfase no encaminhamento, na distribuição e na predição de conteúdos multimídia.

Palavras chave: Comunicações Multimídia; Sistemas Multimídia; Redes Definidas por *Software*.

## ABSTRACT

In the last few years, multimedia applications and services, such as for multimedia sharing services for audio and video contents, and for online gaming, among others, are becoming more and more popular. On the other hand, these multimedia applications and services increase the network traffic load and make the traffic profiles more detailed and complex, requiring the use of specific approaches for multimedia communications. In this scenario, this work proposes and explores, as resources of a multimedia gateway in the context of OpenFlow Software-Defined Networks (SDN), some approaches and resources that aim to enhance multimedia communications over the network, with focus on multimedia content delivery, multimedia distribution and multimedia content prediction. For this purpose, in the context of a multimedia gateway in OpenFlow SDN, the following aspects are especially considered and discussed in this work: (I) identification, classification and differentiated routing of multimedia traffic flows; (II) per-flow routing with Quality of Service (QoS) support; (III) multimedia content caching; and, (IV) multimedia content prediction. In this context, for experimentation and evaluation purposes of the approaches proposed in this work, an emulation environment and a simulation environment are used to obtain experimental results. In summary, the obtained results show that the approaches proposed in this work, as resources of a multimedia gateway in OpenFlow SDN, are able to provide resources to enhance multimedia communications over the network, in this case, with emphasis on multimedia content delivery, multimedia distribution and multimedia content prediction.

**Keywords:** Multimedia Communications; Multimedia Systems; Software-Defined Networking.



## LISTA DE ILUSTRAÇÕES

Figura 1. Tráfego IP global por categoria de aplicação. Dados de 2016 e projeções para 2021. Fonte: Cisco Systems (2017b).....	20
Figura 2. Passos realizados pelo algoritmo discreto do filtro Kalman. Fonte: Adaptado de Oliveira (2011). .....	38
Figura 3. Passos realizados pelo algoritmo discreto do filtro Kalman e pelo subfiltro de Kalman. Fonte: Adaptado de Oliveira (2011). .....	41
Figura 4. Arquitetura SDN e suas abstrações fundamentais. Fonte: Kreutz et al. (2015). .....	42
Figura 5. Decisões de encaminhamento baseadas em fluxos em dispositivos SDN OpenFlow. Fonte: Kreutz et al. (2015).....	45
Figura 6. <i>Gateway</i> multimídia em ambientes SDN OpenFlow. ....	55
Figura 7. Sistemas transmissor e receptor multimídia, com o <i>gateway</i> multimídia, em um ambiente SDN OpenFlow.....	56
Figura 8. <i>Match fields</i> e estatísticas passíveis de utilização de acordo com as versões 1.0 até 1.4 do protocolo OpenFlow. Fonte: Kreutz et al. (2015). .....	57
Figura 9. Possível organização de <i>bits</i> para a identificação e classificação dos serviços multimídia de dados, de áudio, de voz e de vídeo na rede, bem como dos canais de comunicação associados aos mesmos, ambos por meio do campo DSCP do protocolo IP. ....	59
Figura 10. Visão geral da arquitetura de alto nível do <i>gateway</i> multimídia no ambiente SDN OpenFlow. ....	60
Figura 11. Visão geral quanto à oferta de diferentes níveis de QoS junto ao <i>gateway</i> multimídia: abordagem baseada no emprego de múltiplas classes de tráfego, com definições de alocação de banda e de priorização distintas entre si, e classificações de tráfego por meio de identificadores de serviços multimídia. ....	63
Figura 12. Visão geral quanto à oferta de recursos de cache junto ao <i>gateway</i> multimídia: conteúdo multimídia recuperado e distribuído diretamente do cache aos clientes da rede (cenário A) e conteúdo multimídia recuperado do servidor multimídia na Internet, armazenado em cache e distribuído aos clientes da rede (cenário B). ....	64
Figura 13. Visão geral da estrutura funcional do módulo preditor de conteúdos multimídia, nesse caso, baseado em identificadores de serviços multimídia e fornecido por meio do filtro de Kalman e do subfiltro de Kalman. ....	65
Figura 14. Sistemas transmissor e receptor do conteúdo multimídia, com o adaptador de tráfego multimídia e o <i>gateway</i> multimídia, em um ambiente SDN OpenFlow. ....	67
Figura 15. Ambiente de emulação utilizado para a obtenção de resultados experimentais: Sistemas transmissor e receptor dos fluxos de tráfego multimídia de áudio, de voz, de vídeo e de dados, roteador R1, controlador SDN e o <i>gateway</i> multimídia, ambos em um cenário SDN OpenFlow. ....	73
Figura 16. Topologia experimental do ambiente de emulação: <i>Switches</i> S1, S2 e S3 e suas interfaces de conexão com os <i>hosts</i> da rede. ....	73
Figura 17. Saltos ( <i>hops</i> ) entre um dos clientes multimídia ( <i>host</i> H7) e um dos servidores multimídia ( <i>host</i> H2) no ambiente de emulação.....	75
Figura 18. Exemplo de definições para a rotulação de dois fluxos experimentados no ambiente de emulação: fluxos de dados pertinentes ao serviço de hospedagem <i>web</i> (via HTTP) e fluxos de voz pertinentes ao canal 1 do gerador de tráfego multimídia. ....	77
Figura 19. Exemplo de fluxos de tráfego multimídia de dados sobre HTTP e seus respectivos identificadores DSCP no ambiente de emulação.....	79

Figura 20. Exemplo de fluxos de tráfego multimídia de voz pertinentes ao canal 1 do gerador de tráfego multimídia e seus respectivos identificadores DSCP no ambiente de emulação. ...	79
Figura 21. Exemplo de definições para o encaminhamento diferenciado de dois fluxos experimentados no ambiente de emulação: fluxos de dados pertinentes ao canal 1 do gerador de tráfego multimídia e fluxos de voz pertinentes ao canal 2 do gerador de tráfego multimídia. ....	82
Figura 22. Exemplo de definições complementares para o encaminhamento diferenciado dos fluxos de dados pertinentes ao canal 1 do gerador de tráfego multimídia no ambiente de emulação. ....	84
Figura 23. Exemplo de recebimento (A) e encaminhamento diferenciado (B) de fluxos de tráfego multimídia de dados (canal 1) junto ao <i>gateway</i> multimídia no ambiente de emulação. ....	85
Figura 24. Exemplo de recebimento (A) e encaminhamento diferenciado (B) de fluxos de tráfego multimídia de voz (canal 2) junto ao <i>gateway</i> multimídia no ambiente de emulação. ....	86
Figura 25. Exemplo de recebimento de fluxos multimídia de dados, de áudio, de voz e de vídeo pelo cliente multimídia H6 no ambiente de emulação, ambos gerados por meio do gerador de tráfego multimídia no canal 1. ....	87
Figura 26. Topologia experimental para as avaliações sobre as características de tráfego, tempos de transmissão, variações do atraso e perda de pacotes no ambiente de emulação. ....	90
Figura 27. Saltos ( <i>hops</i> ) entre o cliente multimídia H6 e o servidor multimídia H2 (A) e entre o cliente multimídia H7 e o servidor multimídia H2 (B), ambos utilizados nas experimentações com o gerador de tráfego multimídia. ....	90
Figura 28. Saltos ( <i>hops</i> ) entre o cliente multimídia H4 e o servidor multimídia H1 (A) e entre o cliente multimídia H3 e o servidor multimídia H1 (B), ambos utilizados nas experimentações com o iPerf. ....	91
Figura 29. Histogramas pertinentes aos fluxos de áudio, de voz e de vídeo, gerados por meio do gerador de tráfego multimídia e modelados por meio da função de distribuição exponencial, sem o intermédio do <i>gateway</i> multimídia (à esquerda) e com o intermédio do <i>gateway</i> multimídia (à direita), em transmissões de 30 minutos. ....	92
Figura 30. Histogramas pertinentes aos fluxos de áudio, de voz e de vídeo, gerados por meio do gerador de tráfego multimídia e modelados por meio da função de distribuição exponencial, sem o intermédio do <i>gateway</i> multimídia (à esquerda) e com o intermédio do <i>gateway</i> multimídia (à direita), em transmissões de 1 hora. ....	93
Figura 31. Histogramas pertinentes aos fluxos de áudio, de voz e de vídeo, gerados por meio do gerador de tráfego multimídia e modelados por meio da função de distribuição exponencial, sem o intermédio do <i>gateway</i> multimídia (à esquerda) e com o intermédio do <i>gateway</i> multimídia (à direita), em transmissões de 3 horas. ....	94
Figura 32. Histogramas pertinentes aos fluxos de áudio, de voz e de vídeo, gerados por meio do gerador de tráfego multimídia e modelados por meio da função de distribuição exponencial, sem o intermédio do <i>gateway</i> multimídia (à esquerda) e com o intermédio do <i>gateway</i> multimídia (à direita), em transmissões de 6 horas. ....	95
Figura 33. Trecho de <i>log</i> junto ao servidor multimídia H1 ilustrando a ocorrência de perdas de pacotes nos tempos de transmissão de 608-609 segundos e de 1210-1211 segundos. ....	98
Figura 34. Requisições ARP realizadas pelo <i>host</i> cliente H3 em dois momentos (A e B) com perdas de pacotes observadas no ambiente de emulação. ....	99
Figura 35. Utilização dos recursos de rede em duas transmissões simultâneas pertinentes aos conteúdos multimídia de áudio, de vídeo e de dados no ambiente de emulação, nesse caso, sem o emprego de recursos de QoS junto ao <i>gateway</i> multimídia (A) e com emprego de recursos de QoS junto ao <i>gateway</i> multimídia (B). ....	106

Figura 36. Exemplo de dois momentos pertinentes à transmissão do <i>streaming</i> de vídeo no <i>player</i> do cliente multimídia, sem o emprego de recursos de QoS e com a ocorrência de <i>delays</i> pertinentes à <i>bufferizações</i> (A) e com o emprego de recursos de QoS e sem a ocorrência de tais <i>delays</i> (B).	108
Figura 37. <i>Frame</i> de número 563 pertinente ao vídeo de referência original (A) e o <i>frame</i> de mesmo número recebido pelo cliente final com o emprego de recursos de QoS (B) e sem o emprego de recursos de QoS (C).	109
Figura 38. PSNR pertinente ao <i>streaming</i> de vídeo recebido pelo cliente final sem o emprego de recursos de QoS (linha em vermelho) e com o emprego de recursos de QoS (linha em verde) junto ao <i>gateway</i> multimídia.	110
Figura 39. Utilização dos recursos de rede em duas transmissões simultâneas de conteúdos multimídia de vídeo no ambiente de emulação, nesse caso, com o emprego de recursos de QoS junto ao <i>gateway</i> multimídia e por meio de classes de tráfego de maior prioridade (A) e de menor prioridade (B).	112
Figura 40. Utilização dos recursos de rede em duas transmissões simultâneas de conteúdos multimídia de dados no ambiente de emulação, nesse caso, com o emprego de recursos de QoS junto ao <i>gateway</i> multimídia e por meio de classes de tráfego de maior prioridade (A) e de menor prioridade (B).	113
Figura 41. Exemplo de um dos tempos de transmissão para o conteúdo multimídia de dados sobre HTTP: observações realizadas no âmbito da rede (A) e no âmbito da aplicação multimídia (B).	117
Figura 42. Exemplo de um dos tempos de transmissão para o conteúdo multimídia de vídeo sobre HTTP: observações realizadas no âmbito da rede (A) e no âmbito da aplicação multimídia (B).	118
Figura 43. Exemplo de um dos tempos de transmissão para o conteúdo multimídia de áudio sobre HTTP: observações realizadas no âmbito da rede (A) e no âmbito da aplicação multimídia (B).	119
Figura 44. Utilização de recursos de cache junto ao <i>gateway</i> multimídia no primeiro acesso aos conteúdos multimídia de áudio, de vídeo e de dados no ambiente de emulação.	122
Figura 45. Utilização de recursos de cache junto ao <i>gateway</i> multimídia no segundo acesso (A) e no quinto acesso (B) aos mesmos conteúdos multimídia de áudio, de vídeo e de dados no ambiente de emulação.	123
Figura 46. Taxas de transmissão observadas em duas transmissões individuais (A) e em duas transmissões simultâneas (B), tidas como T1 e T2, quanto aos conteúdos multimídia de dados, de vídeo e de áudio, respectivamente, no ambiente de emulação, ambas sem o emprego de recursos de cache junto ao <i>gateway</i> multimídia.	124
Figura 47. Taxas de transmissão observadas nas duas primeiras transmissões individuais (A) e nas duas primeiras transmissões simultâneas (B), tidas como T1 e T2, quanto aos conteúdos multimídia de dados, de vídeo e de áudio, respectivamente, no ambiente de emulação, ambas com o emprego de recursos de cache junto ao <i>gateway</i> multimídia.	125
Figura 48. Mensagens ICMP (via “ <i>ping</i> ”) entre o cliente multimídia H3 e o servidor multimídia H1 sem a inserção de perdas de pacotes (A) e com a inserção de perdas de pacotes na ordem de 10% (B) na interface “Internet” do roteador R1.	127
Figura 49. Mensagens ICMP (via “ <i>ping</i> ”) entre o cliente multimídia H3 e o servidor multimídia H1 sem a inserção de <i>delays</i> de transmissão (A) e com a inserção de <i>delays</i> de transmissão de 200ms (B) na interface “Internet” do roteador R1.	127
Figura 50. Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no primeiro acesso ao conteúdo multimídia de vídeo, sobre HTTP, sem o emprego de recursos de cache na rede local, no cenário com perdas de pacotes na ordem de 10%.	128

Figura 51. Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no segundo acesso ao conteúdo multimídia de vídeo, sobre HTTP, sem o emprego de recursos de cache na rede local, no cenário com perdas de pacotes na ordem de 10%. .....	129
Figura 52. Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no primeiro acesso ao conteúdo multimídia de vídeo, sobre HTTP, com o emprego de recursos de cache junto ao <i>gateway</i> multimídia na rede local, no cenário com perdas de pacotes na ordem de 10%. .....	130
Figura 53. Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no segundo acesso ao conteúdo multimídia de vídeo, sobre HTTP, com o emprego de recursos de cache junto ao <i>gateway</i> multimídia na rede local, no cenário com perdas de pacotes na ordem de 10%. .....	131
Figura 54. Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no primeiro acesso ao conteúdo multimídia de vídeo, sobre HTTP, sem o emprego de recursos de cache na rede local, no cenário com <i>delays</i> de transmissão na ordem de 200ms. ....	132
Figura 55. Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no segundo acesso ao conteúdo multimídia de vídeo, sobre HTTP, sem o emprego de recursos de cache na rede local, no cenário com <i>delays</i> de transmissão na ordem de 200ms. ....	133
Figura 56. Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no primeiro acesso ao conteúdo multimídia de vídeo, sobre HTTP, com o emprego de recursos de cache junto ao <i>gateway</i> multimídia na rede local, no cenário com <i>delays</i> de transmissão na ordem de 200ms. ....	134
Figura 57. Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no segundo acesso ao conteúdo multimídia de vídeo, sobre HTTP, com o emprego de recursos de cache junto ao <i>gateway</i> multimídia na rede local, no cenário com <i>delays</i> de transmissão na ordem de 200ms. ....	135
Figura 58. Taxas de transmissão observadas em 5 requisições ao mesmo conteúdo multimídia de vídeo, sem o emprego de recursos de cache (A) e com o emprego de recursos de cache (B), ambas sem restrições de banda junto ao roteador R1 no ambiente de emulação. Em tal experimentação, a estrutura de cache foi limpa e recriada após cada requisição ao conteúdo de vídeo, de modo que cada requisição fosse tida como uma “nova requisição inicial”. .....	137
Figura 59. Identificadores de serviços multimídia observados e preditos no ambiente de simulação, com predições realizadas a partir de observações em dois testes distintos (A e B), ambos considerando 48 fluxos simulados aleatoriamente para quatro serviços multimídia e com um canal de comunicação para cada serviço multimídia em questão. ....	142
Figura 60. Identificadores de serviços multimídia observados e preditos no ambiente de simulação, com predições realizadas a partir de observações em dois testes distintos (A e B), ambos considerando 48 fluxos simulados aleatoriamente para quatro serviços multimídia, com quatro canais de comunicação para um dos serviços multimídia, três canais de comunicação para dois dos serviços multimídia e dois canais de comunicação para um dos serviços multimídia. ....	144
Figura 61. Definições empregadas para viabilizar a rotulação de fluxos de dados pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação. ....	159
Figura 62. Definições empregadas para viabilizar a rotulação de fluxos de dados pertinentes ao serviço de hospedagem <i>web</i> sobre HTTP no ambiente de emulação. ....	161
Figura 63. Definições empregadas para viabilizar a rotulação de fluxos de dados pertinentes ao serviço de transferência de arquivos sobre FTP no ambiente de emulação. ....	161
Figura 64. Definições empregadas para viabilizar a rotulação de fluxos de áudio pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação. ....	162
Figura 65. Definições empregadas para viabilizar a rotulação de fluxos de áudio pertinentes ao serviço de <i>streaming</i> de áudio sobre RTP no ambiente de emulação. ....	163

Figura 66. Definições empregadas para viabilizar a rotulação de fluxos de áudio pertinentes ao serviço de <i>streaming</i> de áudio sobre RTSP no ambiente de emulação. ....	164
Figura 67. Definições empregadas para viabilizar a rotulação de fluxos de voz pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação. ....	165
Figura 68. Definições empregadas para viabilizar a rotulação de fluxos de vídeo pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação. ....	166
Figura 69. Definições empregadas para viabilizar a rotulação de fluxos de vídeo pertinentes ao serviço de <i>streaming</i> de vídeo sobre RTP no ambiente de emulação. ....	167
Figura 70. Definições empregadas para viabilizar a rotulação de fluxos de vídeo pertinentes ao serviço de <i>streaming</i> de vídeo sobre RTSP no ambiente de emulação. ....	168
Figura 71. Fluxos de tráfego multimídia de dados nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, e seus respectivos identificadores DSCP no ambiente de emulação. ....	169
Figura 72. Fluxos de tráfego multimídia de dados sobre HTTP (A) e sobre FTP (B), pertinentes aos serviços de hospedagem <i>web</i> e de transferência de arquivos, respectivamente, e seus respectivos identificadores DSCP no ambiente de emulação. ....	170
Figura 73. Fluxos de tráfego multimídia de áudio nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, e seus respectivos identificadores DSCP no ambiente de emulação. ....	171
Figura 74. Fluxos de tráfego multimídia de áudio sobre RTSP (A) e sobre RTP (B), pertinentes ao serviço de <i>streaming</i> de áudio, e seus respectivos identificadores DSCP no ambiente de emulação. ....	172
Figura 75. Fluxos de tráfego multimídia de voz nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, e seus respectivos identificadores DSCP no ambiente de emulação. ....	173
Figura 76. Fluxos de tráfego multimídia de vídeo nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, e seus respectivos identificadores DSCP no ambiente de emulação. ....	174
Figura 77. Fluxos de tráfego multimídia de vídeo sobre RTSP (A) e sobre RTP (B), pertinentes ao serviço de <i>streaming</i> de vídeo, e seus respectivos identificadores DSCP no ambiente de emulação. ....	175
Figura 78. Definições empregadas para viabilizar o encaminhamento diferenciado dos fluxos de dados pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação. ....	176
Figura 79. Definições complementares empregadas para viabilizar o encaminhamento diferenciado dos fluxos de dados pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação. ....	177
Figura 80. Definições empregadas para viabilizar o encaminhamento diferenciado dos fluxos de áudio pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação. ....	179
Figura 81. Definições empregadas para viabilizar o encaminhamento diferenciado dos fluxos de voz pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação. ....	180
Figura 82. Definições empregadas para viabilizar o encaminhamento diferenciado dos fluxos de vídeo pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação. ....	181
Figura 83. Exemplo de recebimento de fluxos multimídia de dados nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao <i>gateway</i> multimídia no ambiente de emulação. ....	183

Figura 84. Exemplo de encaminhamento diferenciado de fluxos multimídia de dados nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao <i>gateway</i> multimídia no ambiente de emulação. ....	184
Figura 85. Exemplo de recebimento de fluxos multimídia de áudio nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao <i>gateway</i> multimídia no ambiente de emulação. ....	185
Figura 86. Exemplo de encaminhamento diferenciado de fluxos multimídia de áudio nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao <i>gateway</i> multimídia no ambiente de emulação. ....	186
Figura 87. Exemplo de recebimento de fluxos multimídia de voz nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao <i>gateway</i> multimídia no ambiente de emulação. ....	187
Figura 88. Exemplo de encaminhamento diferenciado de fluxos multimídia de voz nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao <i>gateway</i> multimídia no ambiente de emulação. ....	188
Figura 89. Exemplo de recebimento de fluxos multimídia de vídeo nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao <i>gateway</i> multimídia no ambiente de emulação. ....	189
Figura 90. Exemplo de encaminhamento diferenciado de fluxos multimídia de vídeo nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao <i>gateway</i> multimídia no ambiente de emulação. ....	190
Figura 91. Exemplo de recebimento de fluxos multimídia de áudio, de voz, de vídeo e de dados, ambos gerados por meio do gerador de tráfego multimídia, pelos clientes multimídia H6 (canal 1, A) e H7 (canal 2, B) no ambiente de emulação. ....	191

## LISTA DE TABELAS

Tabela 1. Possíveis identificadores DSCP para os serviços multimídia de dados, de áudio, de voz e de vídeo na rede, bem como de seus respectivos canais de comunicação. ....	59
Tabela 2. Configurações de endereçamento IP atribuídas aos <i>hosts</i> do ambiente de emulação. ....	74
Tabela 3. Fluxos de tráfego multimídia e seus respectivos identificadores de serviços multimídia, por meio de rótulos DSCP, experimentados no ambiente de emulação. ....	76
Tabela 4. Síntese de configurações realizadas no âmbito das aplicações transmissoras e receptoras do gerador de tráfego multimídia quanto aos fluxos de dados, de áudio, de voz e de vídeo experimentados no ambiente de emulação. ....	80
Tabela 5. Tempos de transmissão para os fluxos de tráfego gerados por meio do iPerf, sobre o protocolo UDP, sem o intermédio do <i>gateway</i> multimídia e com o intermédio do <i>gateway</i> multimídia. ....	96
Tabela 6. Variações do atraso ( <i>jitter</i> ) para os fluxos de tráfego gerados por meio do iPerf, sobre o protocolo UDP, sem o intermédio do <i>gateway</i> multimídia e com o intermédio do <i>gateway</i> multimídia. ....	96
Tabela 7. Percentuais de datagramas perdidos para os fluxos de tráfego gerados por meio do iPerf, sobre o protocolo UDP, sem o intermédio do <i>gateway</i> multimídia e com o intermédio do <i>gateway</i> multimídia. ....	96
Tabela 8. Taxas de transmissão para os fluxos pertinentes aos conteúdos multimídia de áudio, de vídeo e de dados, sem o emprego de recursos de QoS junto ao <i>gateway</i> multimídia, em transmissões individuais e em transmissões simultâneas de ambos no ambiente de emulação. ....	102
Tabela 9. Taxas de transmissão para os fluxos pertinentes aos conteúdos multimídia de áudio, de vídeo e de dados, com o emprego de recursos de QoS junto ao <i>gateway</i> multimídia, em transmissões individuais e em transmissões simultâneas de ambos no ambiente de emulação. ....	104
Tabela 10. Tempos de transmissão para os conteúdos multimídia de áudio, de vídeo e de dados, em transmissões individuais, sem o emprego de recursos de cache e com o emprego de recursos de cache junto ao <i>gateway</i> multimídia. ....	120
Tabela 11. Tempos de transmissão para os conteúdos multimídia de áudio, de vídeo e de dados, em transmissões simultâneas, sem o emprego de recursos de cache e com o emprego de recursos de cache junto ao <i>gateway</i> multimídia. ....	120
Tabela 12. Mapeamento de números aleatórios gerados no ambiente de simulação, nesse caso, representando os identificadores de serviços multimídia pertinentes aos fluxos de tráfego dos serviços de dados, de áudio, de voz e de vídeo e seus respectivos canais de comunicação, de acordo com os DSCPs empregados no ambiente de emulação. ....	140
Tabela 13. Exemplo dos 30 primeiros números aleatórios gerados/observados no âmbito dos identificadores de serviços multimídia simulados no ambiente computacional e seus respectivos mapeamentos quanto aos identificadores DSCPs ilustrados no painel superior (A) da Figura 59. ....	143
Tabela 14. Exemplo dos 30 primeiros números aleatórios gerados/observados no âmbito dos identificadores de serviços multimídia simulados no ambiente computacional e seus respectivos mapeamentos quanto aos identificadores DSCPs ilustrados no painel superior (A) da Figura 60. ....	145

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
ARP	Address Resolution Protocol
AS	Autonomous System
CaaS	Caching as a Service
CBQ	Class Based Queueing
CDN	Content Distribution Network
CIDR	Classless Interdomain Routing
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DVMRP	Distance Vector Multicast Routing Protocol
FTP	File Transfer Protocol
GB	Gigabyte
GSM	Global System for Mobile Communications
HTB	Hierarchical Token Bucket
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IGMP	Internet Group Management Protocol
IoT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
LAN	Local Area Network
MAC	Media Access Control
MB	Megabyte
MMS	Microsoft Media Service
MOS	Mean Opinion Score
MOSPF	Multicast Open Shortest Path First
MPEG	Moving Picture Experts Group
MPLS	Multiprotocol Label Switching
MTG	Multimedia Traffic Generator



NAT	Network Address Translation
NI	Northbound Interface
NOS	Network Operating System
NSIS	Next Steps in Signaling
PCP	Priority Code Point
PIM	Protocol Independent Multicast
POF	Protocol Oblivious Forwarding
PSNR	Peak Signal-to-Noise Ratio
PTT	Ponto de Troca de Tráfego
QoE	Quality of Experience
QoS	Quality of Service
RED	Randomly Early Detection
RFC	Request for Comments
RPE	Resource Prediction Engine
RSVP	Resource Reservation Protocol
RTCP	Real-Time Transport Control Protocol
RTP	Real-Time Transport Protocol
RTSP	Real Time Streaming Protocol
SDN	Software-Defined Network
SFQ	Stochastic Fairness Queuing
SI	Southbound Interface
SIP	Session Initiation Protocol
TC	Traffic Class
TCP	Transmission Control Protocol
ToS	Type of Service
UDP	User Datagram Protocol
VANET	Veicular Ad Hoc Network
VoD	Video-on-Demand
VoIP	Voice over IP
WMSN	Wireless Multimedia Sensor Network

## SUMÁRIO

1 – Introdução.....	20
1.1 – Motivações.....	22
1.2 – Objetivos Gerais e Específicos .....	24
1.3 – Contribuições da Tese .....	25
1.4 – Organização da Tese.....	27
2 – Revisão Bibliográfica .....	29
2.1 – Comunicação Multimídia .....	29
2.1.1 – Aplicações Multimídia .....	30
2.1.2 – Protocolos Multimídia .....	32
2.1.3 – Distribuição de Conteúdos Multimídia na Internet .....	33
2.2 – Modelos de Predição .....	34
2.2.1 – Filtro de Kalman.....	35
2.2.2 – Subfiltro de Kalman.....	38
2.3 – Redes Definidas por <i>Software</i> .....	41
2.3.1 – Arquitetura, Principais Componentes e Princípios Básicos .....	42
2.3.2 – Protocolo OpenFlow.....	46
2.4 – Trabalhos Relacionados.....	48
2.5 – Síntese e Encerramento do Capítulo.....	53
3 – <i>Gateway</i> Multimídia em Ambientes SDN OpenFlow.....	54
3.1 – Visão Geral.....	54
3.2 – Identificação e Classificação de Fluxos de Tráfego Multimídia .....	57
3.3 – Arquitetura e Recursos Funcionais.....	60
3.4 – Adaptador de Tráfego Multimídia.....	66
3.5 – Algumas Considerações .....	68
3.6 – Síntese e Encerramento do Capítulo.....	69
4 – Experimentos e Resultados.....	70
4.1 – Ambientes e Ferramentas de Experimentação e Avaliação.....	70
4.2 – Resultados e Discussão.....	76
4.2.1 – Rotulação de Fluxos de Tráfego Multimídia.....	76
4.2.2 – Identificação e Encaminhamento Diferenciado de Fluxos de Tráfego Multimídia .....	80
4.2.3 – Características de Tráfego, Tempos de Transmissão, Variações do Atraso e Perda de Pacotes .....	88
4.2.4 – Encaminhamento por Fluxos com Suporte à QoS.....	100
4.2.5 – Cache de Conteúdos Multimídia .....	114

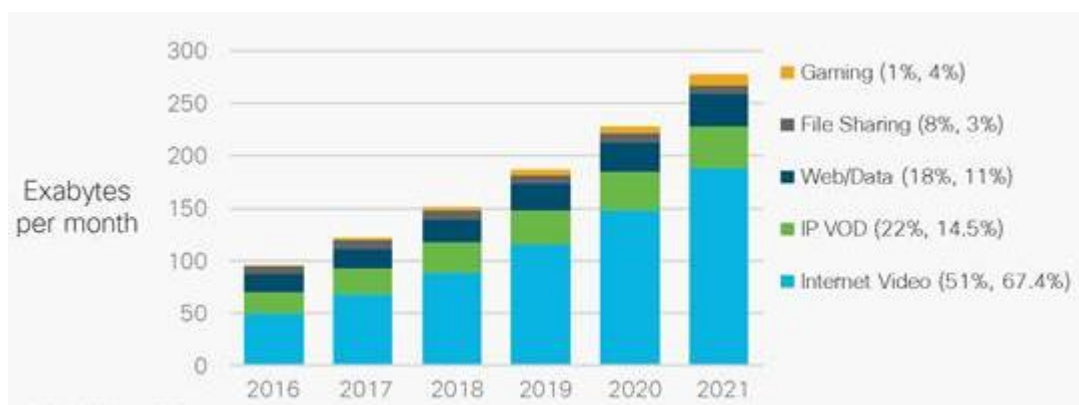
4.2.6 – Predição de Conteúdos Multimídia .....	139
4.3 – Síntese e Encerramento do Capítulo.....	146
5 – Conclusão e Trabalhos Futuros .....	147
REFERÊNCIAS .....	152
Apêndice A – Regras de Fluxos .....	159
A.1 – Rotulação de Fluxos de Tráfego Multimídia .....	159
A.2 – Identificação e Encaminhamento Diferenciado de Fluxos de Tráfego Multimídia..	176

## 1 – Introdução

Nos últimos anos, foi possível observar e vivenciar uma contínua evolução quanto aos serviços e aplicações fornecidas e exploradas no âmbito da rede (Internet). Dentre esses serviços e aplicações, aqueles com ênfase em conteúdos multimídia, tais como para o acesso e compartilhamento de conteúdos de áudio e vídeo, para *gaming* ou para realização de telefonia e videoconferência pela Internet, dentre outros, tornaram-se cada vez mais populares. Como exemplo, de acordo com relatórios técnicos disponibilizados pela empresa Cisco Systems (CISCO SYSTEMS, 2017a; CISCO SYSTEMS, 2017b), 73% de todo o tráfego IP global gerado na Internet em 2016 foi referente a tráfego de vídeo sobre IP e 1% referente a tráfego de *gaming*, com projeções de que esses percentuais saltem para 82% e 4%, respectivamente, no ano de 2021 (Figura 1). Em termos de volume tráfego, esses percentuais equivalem aos totais próximos de 900 Exabytes realizados no ano de 2016 (dos 1,2 Zettabytes totais naquele ano) e de projeções próximas de 2,8 Zettabytes para o ano de 2021 (dos 3,3 Zettabytes previstos para tal ano). De acordo com tais projeções, em que o tráfego IP global irá aumentar quase 3 vezes nos próximos 5 anos e quase 127 vezes entre 2005 e 2021, seria necessário mais de 5 milhões de anos para que um indivíduo pudesse assistir a quantidade de conteúdos de vídeo que cruzará mensalmente as redes IP globais em 2021, com cerca de 1 milhão de minutos de conteúdos de vídeo sendo trafegados por segundo na rede até tal ano (CISCO SYSTEMS, 2017a; CISCO SYSTEMS, 2017b).

**Figura 1.** Tráfego IP global por categoria de aplicação. Dados de 2016 e projeções para 2021.

Fonte: Cisco Systems (2017b).



Diante desse cenário, esses novos serviços e aplicações tornaram o fluxo de tráfego de rede, além de mais intenso, mais detalhado e complexo, uma vez que seus requisitos e particularidades encontram uma série de dificuldades e limitações que vão desde seu projeto e desenvolvimento até a forma com as quais seus conteúdos são distribuídos, acessados e até percebidos, em termos de QoE (*Quality of Experience*), por exemplo, pelos usuários finais ao longo da rede (KUROSE; ROSS, 2010; TANENBAUM, 2003; MORALES; BERGSTROM, 2017; ROBITZA et al., 2017). Como resultado, uma série de trabalhos e pesquisas acadêmicas discorre a respeito de soluções e alternativas de modo a contribuir com esse cenário, tais como para a entrega e distribuição de conteúdos multimídia (CIUBOTARU; MUNTEAN; MUNTEAN, 2017; HERBAUT et al., 2017; FU; LIU; ZHU, 2017), para a predição e provisionamento de recursos multimídia (KRYFTIS et al., 2016a; KRYFTIS et al., 2016b; SUN et al., 2017), para a indexação, recuperação e classificação multimídia (SAFADI; DERBAS; QUÉNOT, 2015; ZHAI et al., 2017; LIU; GUO; AN, 2017) e para recursos de QoS (*Quality of Service*) e de QoE no âmbito de comunicações multimídia (KUCMINSKI et al., 2017; PEDERSEN; DEY, 2016; MAIA; YEHIA; ERRICO, 2015), dentre outros.

Nesse cenário de rede, a atual arquitetura Internet, tida como “ossificada” e com uma série de deficiências e limitações (de escalabilidade, de segurança e de mobilidade, por exemplo), embora usual e funcional, possui características de evolução e inovação que, além de dificultar tais contribuições, são desfavoráveis para sua manutenção enquanto “Internet do futuro” (MCKEOWN et al., 2008; MOREIRA et al., 2009; PAUL; PAN; JAIN, 2011). Alguns autores, tais como Moreira et al. (2009), por exemplo, definem que a evolução da Internet até hoje tem se dado por meio de “remendos”, tais como pela criação e utilização de sub-redes, Sistemas Autônomos (*Autonomous Systems, AS*), servidores DNS (*Domain Name System*), CIDR (*Classless Interdomain Routing*), NAT (*Network Address Translation*) e modificações em protocolos como TCP e IP, dentre outros, de modo que uma “Nova Internet” se faz necessária para atender aos requisitos atuais e também das futuras aplicações de rede.

Nesse contexto, conforme descrito nos trabalhos de Kreutz et al. (2015), Xia et al. (2015) e Nunes et al. (2014), com uma abordagem centrada na separação dos planos de dados (encaminhamento) e de controle, as Redes Definidas por *Software* (*Software-Defined Network, SDN*) representam um paradigma emergente que objetiva simplificar e flexibilizar o gerenciamento da rede, além de permitir e facilitar sua evolução e inovação.

Em SDN, que possui infraestrutura física similar às redes tradicionais, a inteligência da rede está centralizada em controladores baseados em *software* (plano de controle), com dispositivos simples, e programáveis por meio de uma interface aberta, realizando o encaminhamento de pacotes (plano de dados) (KREUTZ et al., 2015; XIA et al., 2015; NUNES et al., 2014). De modo a objetivar a compatibilidade e interoperabilidade entre os diversos dispositivos dos planos de dados e o controlador, toda a comunicação entre ambos se dá por meio de interfaces abertas e padronizadas de comunicação, algo difícil em redes tradicionais, em especial, devido à grande variedade de interfaces proprietárias (e fechadas) e pela natureza distribuída do plano de controle (KREUTZ et al., 2015). Nesse cenário, o protocolo OpenFlow (MCKEOWN et al., 2008) tem sido amplamente utilizado para permitir a comunicação entre os dispositivos do plano de dados e o controlador.

Dessa forma, além de possibilitar a consolidação dos diversos dispositivos da rede, a arquitetura SDN facilita a implantação de novos protocolos e aplicações, simplifica e flexibiliza o gerenciamento da rede e possibilita múltiplas opções de controle quanto aos seus fluxos de tráfego (KREUTZ et al., 2015; NUNES et al., 2014, MENDIOLA et al., 2017). Características amplamente benéficas e favoráveis à inovação e evolução da rede e de suas aplicações, de modo geral.

Diante desse cenário, esta Tese discorre sobre um *gateway* com recursos multimídia, tido como *gateway* multimídia, enquanto plataforma para aprimorar comunicações multimídia no âmbito de Redes Definidas por *Software* OpenFlow. Nesse contexto, é importante destacar que, nesta Tese, entende-se (e delimita-se) por “aprimorar comunicações multimídia” o fornecimento de recursos e funcionalidades que contribuam para o encaminhamento, para a distribuição e para a predição de conteúdos multimídia na rede, bem como na percepção desses em termos de QoE pelo usuário final.

## **1.1 – Motivações**

Aplicações e serviços multimídia estão dentre as aplicações e serviços de maior popularidade e sucesso na Internet. Há uma grande variedade de soluções que enfatizam, por exemplo, transmissões de áudio, de voz, de vídeo e de dados no contexto multimídia. Por outro lado, essa popularidade e sucesso comprometem e são comprometidas, especialmente, por questões funcionais e arquiteturas da rede e de suas aplicações. Como resultado,

inúmeros esforços da academia e da indústria discorrem sobre comunicações multimídia. Eventos tradicionais da área, tais como o Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia), o *IEEE International Symposium on Multimedia (ISM)* e o *ACM Multimedia*, dentre vários outros, reúnem milhares de participantes em suas edições, demonstrando o quão relevante e abrangente é tal área na Computação.

Nesse cenário, o paradigma de Redes Definidas por *Software* representa uma área de pesquisa capaz de fornecer múltiplas contribuições à rede e suas aplicações. Pesquisas e contribuições recentes em SDN, como as sintetizadas nos trabalhos de Kreutz et al. (2015), Xia et al. (2015) e Nunes et al. (2014), por exemplo, discorrem sobre os benefícios associados a sua estrutura e arquitetura funcional. A separação dos planos de dados e de controle, a flexibilidade de compatibilidade e interoperabilidade por meio de interfaces abertas e padronizadas de comunicação e os recursos de programação e gerência centralizada possibilitam o desenvolvimento de aplicações e *middleboxes* capazes de interagir e controlar a rede de modo impensado no passado (KREUTZ et al., 2015; XIA et al., 2015; NUNES et al., 2014; LARA; KOLASANI; RAMAMURTHY, 2014; GUEDES et al., 2012; MCKEOWN et al., 2008). Possibilidades que fazem das SDNs uma potencial arquitetura para a “Internet do Futuro” (HAKIRI et al., 2014; GUEDES et al., 2012), concentrando esforços da academia e também da indústria em seu estudo e disseminação.

Como exemplo, enquanto esforços da academia, entidades como IEEE<sup>1</sup> e ACM<sup>2</sup> possuem eventos específicos voltados para SDN, tais como o *IEEE Conference on Network Functions Virtualization and Software Defined Networking (IEEE NFV-SDN)* e o *ACM SIGCOMM Symposium on SDN Research (SOSR)*, dentre outros. Na indústria, a *Open Networking Foundation (ONF)*<sup>3</sup>, uma entidade destinada a divulgar e promover a adoção de SDN, possui grandes empresas como membros, tais como AT&T, Cisco, Google e Microsoft, dentre outras. Em parceria, membros da indústria e da academia se reúnem anualmente no *Open Networking Summit*<sup>4</sup> para, também, divulgar e promover a adoção de SDN.

O resultado desses esforços é traduzido, pela academia e pela indústria, em uma série de publicações associadas ao tema e, também, em soluções de mercado em SDN e no protocolo OpenFlow. Essas contribuições, suas aplicabilidades e o retorno de suas experiências, tais como as divulgadas pelos engenheiros e pesquisadores do Google em Yap

<sup>1</sup> *Institute of Electrical and Electronics Engineers (IEEE)* em <https://www.ieee.org>

<sup>2</sup> *Association for Computing Machinery (ACM)* em <https://www.acm.org>

<sup>3</sup> *Open Networking Foundation (ONF)* em <https://www.opennetworking.org>

<sup>4</sup> *Open Networking Summit* em <http://opennetsummit.org>

et al. (2017) e Jain et al. (2013), por exemplo, bem como aquelas já mencionadas anteriormente, sintetizadas nos trabalhos de Kreutz et al. (2015), Xia et al. (2015), Nunes et al. (2014) e Lara, Kolasani e Ramamurthy (2014), dentre outros, permitem observar o quão relevante e abrangente é esse tema em Computação.

Diante desse cenário, as principais motivações para esta Tese podem ser resumidas:

- (I) Pelas altas demandas por aplicações e serviços multimídia e seus desafios quanto à comunicação ao longo da rede;
- (II) Pelo potencial de inovação e evolução para a rede e suas aplicações no âmbito de SDNs e do protocolo OpenFlow;
- (III) Pelos esforços e contribuições da academia e da indústria em divulgar e promover a adoção de SDN, inclusive enquanto possível arquitetura para a “Internet do Futuro”;
- (IV) Pela possibilidade de se contribuir com esse cenário, por meio de um *gateway* multimídia no âmbito de SDNs OpenFlow, com ênfase no encaminhamento, na distribuição e na predição de conteúdos multimídia, bem como na percepção desses em termos de QoE pelo usuário final.

## 1.2 – Objetivos Gerais e Específicos

Esta Tese tem como objetivo geral discorrer e avaliar o emprego de um *gateway* com recursos multimídia, tido como *gateway* multimídia, enquanto plataforma para aprimorar comunicações multimídia no âmbito de Redes Definidas por *Software* OpenFlow. Nesse contexto, é importante destacar que, nesta Tese, entende-se (e delimita-se) por “aprimorar comunicações multimídia” o fornecimento de recursos e funcionalidades que contribuam para o encaminhamento, para a distribuição e para a predição de conteúdos multimídia na rede, bem como na percepção desses em termos de QoE pelo usuário final.

Dessa forma, no âmbito de um *gateway* multimídia em SDN OpenFlow, são objetivos específicos desta Tese:



- (I) Viabilizar um método para identificação, classificação e encaminhamento diferenciado de fluxos de tráfego multimídia na rede, em ambos os casos, por meio de identificadores de serviços multimídia;
- (II) Viabilizar um método para a oferta de diferentes níveis de QoS para os fluxos de tráfego multimídia da rede, em especial, por meio de múltiplas classes de tráfego com distintas configurações em termos de alocação de banda e de priorização, bem como considerando os identificadores de serviços multimídia pertinentes a cada fluxo de tráfego em questão;
- (III) Viabilizar um método para aproximar o conteúdo multimídia requisitado dos clientes finais, de modo a possibilitar, por exemplo, melhores tempos e taxas de transmissão no acesso e distribuição de conteúdos multimídia na rede;
- (IV) Viabilizar um método de predição de conteúdos multimídia por meio de identificadores de serviços/conteúdos multimídia, de modo a possibilitar, por exemplo, que futuros conteúdos multimídia sejam previstos na rede.

### 1.3 – Contribuições da Tese

Em linhas gerais, no âmbito de um *gateway* multimídia em SDN OpenFlow e dos objetivos gerais e específicos delineados na Seção anterior, as principais contribuições desta Tese podem ser sintetizadas:

- (I) Em uma abordagem para identificação, classificação e encaminhamento diferenciado de fluxos de tráfego multimídia na rede, por meio de identificadores de serviços multimídia para os serviços de dados, de áudio, de voz e de vídeo, em especial, fornecidos via rótulos DSCP e no âmbito SDN OpenFlow (pertinente ao objetivo específico “I”);
- (II) Em uma abordagem para o encaminhamento por fluxos com suporte à QoS no âmbito SDN OpenFlow, em especial, por meio de múltiplas classes de tráfego, combinando recursos de compartilhamento de banda e de priorização entre classes de tráfego, bem como do emprego de identificadores de serviços multimídia para associação dos diferentes fluxos de tráfego multimídia com suas respectivas classes de tráfego (pertinente ao objetivo específico “II”);

(III) Em uma abordagem baseada no emprego de recursos de Proxy/cache multimídia para aproximar e possibilitar, em especial, melhores tempos e taxas de transmissão no acesso e distribuição de conteúdos multimídia na rede (pertinente ao objetivo específico “III”);

(IV) Em uma abordagem para a predição de conteúdos multimídia, baseada no filtro de Kalman e no subfiltro de Kalman, por meio de identificadores de serviços/conteúdos multimídia (pertinente ao objetivo específico “IV”);

(V) Em uma discussão não exaustiva sobre comunicações multimídia no âmbito de Redes Definidas por *Software* OpenFlow.

Diante desse cenário, é importante destacar que parte dos resultados preliminares obtidos durante a realização desta Tese foi compartilhada com a comunidade acadêmica por meio das seguintes publicações:

1. DIORIO, R. F.; TIMÓTEO, V. S. Multimedia Content Prediction Using the Kalman Filter. In: Proceedings of the Winter Simulation Conference, 2017. **Aceito para publicação (evento em 12/2017).**
2. DIORIO, R. F.; TIMÓTEO, V. S. Multimedia Content Distribution in Software-Defined Networks (SDN). In: Proceedings of the 14th International Conference on Information Systems and Technology Management, p. 2396-2407, 2017. Disponível em: <http://www.contecsi.fea.usp.br/envio/index.php/contecsi/14CONTECSI/paper/view/4668>.
3. DIORIO, R. F.; TIMÓTEO, V. S. Per-Flow Routing with QoS Support to Enhance Multimedia Delivery in OpenFlow SDN. In: Proceedings of the 22º Simpósio Brasileiro de Sistemas Multimídia e Web, p. 167-174, 2016. Disponível em: <http://dl.acm.org/citation.cfm?id=2976844>.
4. DIORIO, R. F.; TIMÓTEO, V. S. Multimedia Content Delivery in OpenFlow SDN: An Approach Based on a Multimedia Gateway. In: Proceedings of the International Symposium on Parallel and Distributed Computing and Computational Science, p. 612-617, 2016. Disponível em: <http://ieeexplore.ieee.org/document/7881414/>.
5. DIORIO, R. F.; TIMÓTEO, V. S. A Platform for Multimedia Traffic Forwarding in Software Defined Networks. In: Proceedings of the 21º Simpósio Brasileiro de Sistemas

Multimídia e Web, p. 177-180, 2015. Disponível em: <http://dl.acm.org/citation.cfm?id=2820464>.

Outras publicações relacionadas ao tema, porém não especificamente no âmbito SDN OpenFlow, são:

6. DIORIO, R. F.; TIMÓTEO, V. S.; URSINI, E. L. Testing an IP-based Multimedia Gateway. *INFOCOMP Journal of Computer Science*, v. 13, p. 21-25, 2014. Disponível em: <http://infocomp.dcc.ufla.br/index.php/INFOCOMP/article/view/19>.
7. DIORIO, R. F.; TIMÓTEO, V. S.; URSINI, E. L. Encaminhamento seletivo de pacotes provido por um Gateway Multimídia baseado em IP. In: *Proceedings of the 12th International Conference on Information Systems and Technology Management*, p. 1-13, 2015. Disponível em: <http://www.contecsi.fea.usp.br/envio/index.php/contecsi/12CONTECSI/paper/view/2797>.

#### 1.4 – Organização da Tese

Esta Tese está organizada em Introdução e outros 4 Capítulos: Revisão Bibliográfica; *Gateway* Multimídia em Ambientes SDN OpenFlow; Experimentos e Resultados; e, Conclusão e Trabalhos Futuros.

Nesse contexto, o Capítulo 2 discorre sobre a revisão bibliográfica, com o objetivo de familiarizar o leitor com os principais conceitos e fundamentos que embasam esta Tese, bem como discutir, de forma não exaustiva, alguns dos trabalhos que estão relacionados com a mesma. Para tal, está organizado em cinco Seções: Comunicação Multimídia; Modelos de Predição; Redes Definidas por *Software*; Trabalhos Relacionados; e, Síntese e Encerramento do Capítulo. Nessa organização, a primeira Seção discorre sobre os conceitos e fundamentos voltados à comunicação multimídia, com ênfase quanto às aplicações multimídia, aos protocolos multimídia e à distribuição de conteúdos multimídia na Internet; a segunda Seção discorre sobre modelos de predição, com ênfase quanto ao filtro de Kalman e ao subfiltro de Kalman; a terceira Seção discorre sobre Redes Definidas por *Software*, com ênfase quanto à arquitetura, principais componentes e princípios básicos das SDNs e quanto

ao protocolo OpenFlow; a quarta Seção discorre sobre alguns dos trabalhos relacionados com esta Tese; e, por sua vez, a quinta Seção aborda uma síntese de encerramento do Capítulo.

O Capítulo 3 discorre sobre o *gateway* multimídia em ambientes SDN OpenFlow. Para tal, está organizado em seis Seções: Visão Geral; Identificação e Classificação de Fluxos de Tráfego Multimídia; Arquitetura e Recursos Funcionais; Adaptador de Tráfego Multimídia; Algumas Considerações; e, Síntese e Encerramento do Capítulo. Nessa organização, a primeira Seção, que trata de uma visão geral sobre o *gateway* multimídia em ambientes SDN OpenFlow, apresenta alguns conceitos e fundamentos quanto ao mesmo em tal cenário de rede; a segunda Seção discorre sobre a abordagem empregada junto ao *gateway* multimídia para a identificação e classificação de fluxos de tráfego multimídia na rede, com ênfase quanto ao emprego de identificadores de serviços multimídia, por meio de rótulos DSCP, para tal finalidade; a terceira Seção discorre sobre a arquitetura e os recursos funcionais propostos para o *gateway* multimídia, com ênfase quanto ao encaminhamento, à distribuição e à predição de conteúdos multimídia; a quarta Seção discorre sobre um elemento intermediário, tido como adaptador de tráfego multimídia, enquanto solução complementar para a rotulação de fluxos de tráfego multimídia na rede; a quinta Seção discorre sobre algumas considerações quanto ao *gateway* multimídia no âmbito de comunicações multimídia; e, por sua vez, a sexta Seção aborda uma síntese de encerramento do Capítulo.

O Capítulo 4 discorre sobre os experimentos e resultados de avaliação. Para tal, está organizado em três Seções: Ambientes e Ferramentas de Experimentação e Avaliação; Resultados e Discussão; e, Síntese e Encerramento do Capítulo. Nessa organização, a primeira Seção apresenta uma visão geral sobre os ambientes e ferramentas de experimentação e avaliação empregadas nesta Tese, nesse caso, no âmbito da emulação e da simulação; a segunda Seção discorre sobre os resultados obtidos por meio de tais ambientes, com ênfase quanto à rotulação de fluxos de tráfego multimídia, à identificação e encaminhamento diferenciado de fluxos de tráfego multimídia, às características de tráfego, tempos de transmissão, variações do atraso e perda de pacotes, ao encaminhamento por fluxos com suporte à QoS, ao cache de conteúdos multimídia e à predição de conteúdos multimídia; e, por sua vez, a terceira Seção aborda uma síntese de encerramento do Capítulo..

Por sua vez, o Capítulo 5 discorre sobre a conclusão e os trabalhos futuros no âmbito desta Tese.

## 2 – Revisão Bibliográfica

Este Capítulo tem o objetivo de familiarizar o leitor com os principais conceitos e fundamentos que embasam esta Tese, bem como discutir, de forma não exaustiva, alguns dos trabalhos que estão relacionados com a mesma. Para tal, a primeira Seção discorre sobre os conceitos e fundamentos voltados à comunicação multimídia, com ênfase quanto às aplicações multimídia, aos protocolos multimídia e à distribuição de conteúdos multimídia na Internet; a segunda Seção discorre sobre modelos de predição, com ênfase quanto ao filtro de Kalman e ao subfiltro de Kalman; a terceira Seção discorre sobre Redes Definidas por *Software*, com ênfase quanto à arquitetura, principais componentes e princípios básicos das SDNs e quanto ao protocolo OpenFlow; e, por sua vez, a quarta Seção discorre sobre alguns dos trabalhos relacionados com esta Tese.

### 2.1 – Comunicação Multimídia

Aplicações e serviços multimídia tornaram-se, nos últimos anos, instrumentos de lazer e trabalho praticamente inseparáveis às rotinas das centenas de milhares de usuários que exploram e utilizam recursos de Internet em escala global em seu dia a dia. Como resultado, o maior percentual do volume de tráfego trocado na Internet na atualidade é pertinente às aplicações e serviços multimídia, em especial, de vídeo (CISCO SYSTEMS, 2017a; CISCO SYSTEMS, 2017b).

Por outro lado, esses novos serviços e aplicações tornaram o fluxo de tráfego de rede, além de mais intenso, mais detalhado e complexo, exigindo o emprego de abordagens específicas no âmbito de comunicações multimídia. Em linhas gerais, essas abordagens compreendem questões que vão desde o projeto e desenvolvimento das aplicações e serviços multimídia até a forma com as quais seus conteúdos são distribuídos, acessados e até percebidos, em termos de QoE, por exemplo, pelos usuários finais ao longo da rede (KUROSE; ROSS, 2010; TANENBAUM, 2003; ROBITZA et al., 2017).

Diante desse cenário, os tópicos seguintes discorrem sobre conceitos e fundamentos voltados à comunicação multimídia, com ênfase quanto às aplicações multimídia, aos protocolos multimídia e à distribuição de conteúdos multimídia na Internet.

### 2.1.1 – Aplicações Multimídia

Comunicações multimídia se dão no âmbito de aplicações multimídia. Por sua vez, é de amplo conhecimento que tais aplicações, também conhecidas como aplicações *stream* ou de tráfego inelástico, em linhas gerais, impõem certas exigências temporais (em especial quanto a atrasos e variações do atraso, *jitter*) e de banda que, além de as distinguirem das demais aplicações de rede, exigem técnicas e abordagens específicas quanto ao seu processo de comunicação.

Nesse contexto, no âmbito do desenvolvimento, Kurose e Ross (2010) discorrem sobre alternativas que podem contribuir em termos de comunicação multimídia, tais como:

- Utilização de algoritmos de compressão para facilitar a transmissão, em termos de largura de banda, de conteúdos de áudio e vídeo, por exemplo;
- Utilização do protocolo UDP, mais simples e com menos controles que o TCP, como protocolo de transporte para alguns conteúdos multimídia;
- Adição de retardos (atrasos) propositais de reprodução e envio de informações redundantes para reduzir, respectivamente, os efeitos da variação do atraso e da perda de pacotes;
- Aquisição antecipada de dados durante a reprodução de alguns conteúdos multimídia armazenados (quando houver capacidade de conteúdo e de largura de banda), etc.

Em linhas gerais, essas considerações objetivam minimizar os efeitos do serviço *best effort* da Internet sobre as aplicações multimídia. Os mesmos autores também discorrem sobre a utilização de algoritmos de compressão de áudio e vídeo, tais como GSM e MPEG, dentre outros, como alternativa às limitações de banda e de armazenamento em termos de seus conteúdos. Quanto à distribuição de conteúdos multimídia, os quais são abordados na Seção 2.1.3, Kurose e Ross (2010) discorrem sobre a utilização de CDNs (*Content Distribution Networks*) como alternativa para aproximar o conteúdo multimídia dos usuários finais e, assim como Tanenbaum (2003), também discorrem sobre o emprego de redes *multicast* de sobreposição como alternativa à redução na carga total de tráfego (em comparação às transmissões *unicast*) na distribuição de conteúdos multimídia ao vivo na Internet.

De modo complementar, alguns trabalhos recentes discorrem sobre a arquitetura e características de plataformas multimídia de vídeo em termos de comunicação multimídia. Como exemplo, no trabalho de Che, Ip e Lin (2015), características quanto aos fluxos de tráfego do portal YouTube<sup>5</sup> são apresentadas de modo a contribuir com questões envolvendo engenharia de tráfego, distribuição de conteúdos multimídia e algoritmos de cache, dentre outros. De modo similar, no trabalho de Adhikari et al. (2015), estudos quanto a arquitetura de aplicação e a estrutura de CDNs dos portais Netflix<sup>6</sup> e Hulu<sup>7</sup> são apresentadas de modo a contribuir no projeto de novos portais e plataformas de *streaming* de vídeo em larga escala.

Questões envolvendo tempos de carregamento de conteúdos, facilidade de utilização, a interatividade e a estética da aplicação e, até mesmo, a influência de fatores e do contexto humano, tais como humor e níveis sociais e de educação, dentre outros, também são fatores observados em termos de comunicação multimídia, nesse caso, com ênfase na QoE percebida pelo usuário final. Como exemplo, no trabalho de Varela et al. (2015), os autores concluem, por meio de análises de três categorias de conteúdos em sites *web*, que a QoE está fortemente correlacionada com a estética, a facilidade de uso e o desempenho (tempo de carregamento de conteúdos) percebidos pelo usuário final. Por sua vez, no trabalho de Zhao, Liu e Chen (2017), os autores sintetizam uma série de contribuições que discorrem, por exemplo, sobre a influência de fatores e do contexto humano na percepção de QoE do usuário final, tais como quanto ao humor e aos níveis sociais e de educação descritos anteriormente, bem como de questões sobre a acuidade visual e auditiva, o gênero, a idade e as condições socioculturais e socioeconômicas dos usuários finais, nesse caso, consideradas no âmbito de transmissões de vídeo no contexto multimídia.

Nesse contexto, mesmo de modo resumido, esses trabalhos permitem observar a importância da aplicação em termos de comunicação multimídia. Perceba que, nesse caso, os desafios que envolvem essa comunicação não estão, necessariamente, relacionados à rede, mas sim à aplicação. As estratégias de desenvolvimento, a arquitetura da aplicação e a forma com a qual os conteúdos multimídia serão acessados e até percebidos pelos usuários finais, dentre outros, são fatores cruciais em tal processo de comunicação. Fatores que podem contribuir ou prejudicar, de modo geral, o processo de comunicação multimídia, nesse caso, no âmbito da aplicação.

---

<sup>5</sup> YouTube em <https://www.youtube.com>

<sup>6</sup> Netflix em <https://www.netflix.com>

<sup>7</sup> Hulu em <http://www.hulu.com>

### 2.1.2 – Protocolos Multimídia

Protocolos são cruciais em todo processo de comunicação ao longo da rede. Em termos de Internet, a qual representa o ambiente e infraestrutura típica para comunicações multimídia, sua estrutura funcional está embasada na pilha de protocolos TCP/IP (KUROSE; ROSS, 2010; TANENBAUM, 2003). Como característica, de um modo ou de outro, o processo envolvendo comunicações multimídia empregará protocolos da pilha TCP/IP como parte de seu processo de comunicação.

Nos níveis de rede e de transporte, protocolos como o IP, TCP e UDP representam a “base” para esse processo de comunicação. Ainda assim, por si só, não fornecem mecanismos suficientes para, por exemplo, fornecer controles quanto à reprodução do conteúdo multimídia pelo cliente ou o gerenciamento de sessões de comunicação envolvendo essas aplicações. Por esse motivo, é comum a utilização (e desenvolvimento) de protocolos específicos para tratar de particularidades voltadas às aplicações multimídia, de modo geral.

Nesse contexto, dentre os diversos protocolos empregados no âmbito de aplicações multimídia e além dos tradicionais IP, TCP e UDP, é comum a utilização de protocolos como RTP (*Real-Time Transport Protocol*), RTCP (*Real-Time Transport Control Protocol*), RTSP (*Real Time Streaming Protocol*) e SIP (*Session Initiation Protocol*), dentre outros. Quando o processo de comunicação envolve reservas de recursos ou transmissões *multicast*, por exemplo, outros protocolos podem ser empregados, tais como RSVP (*Resource Reservation Protocol*), IGMP (*Internet Group Management Protocol*), MOSPF (*Multicast Open Shortest Path First*), DVMRP (*Distance Vector Multicast Routing Protocol*) e PIM (*Protocol Independent Multicast*), dentre outros.

De modo complementar, alguns trabalhos recentes também discorrem sobre propostas para novos protocolos multimídia. Como exemplo, no trabalho de Said et al. (2017), os autores propõem os protocolos IoT-RTP e IoT-RTCP enquanto versões dos protocolos RTP e RTCP, respectivamente, para transmissões multimídia no âmbito de IoT (*Internet of Things*). De modo similar, com ênfase em transmissões de *streaming* de vídeo, os trabalhos de Kwon, Go e Song (2016) e Rezende et al. (2015) discorrem sobre propostas para novos protocolos multimídia no âmbito de redes *wireless* e de VANETs (*Veicular Ad Hoc Networks*), respectivamente. Por sua vez, nos trabalhos de Hasan, Al-Rizzo e Al-Turjman



(2017) e Han et al. (2016), os autores sintetizam uma série de contribuições quanto à protocolos de roteamento no âmbito de WMSNs (*Wireless Multimedia Sensor Networks*).

Nesse contexto, é importante destacar que não é objetivo desta Tese discorrer sobre esses protocolos, mas possibilitar que o leitor perceba que, enquanto protocolos, há uma grande variedade de opções no âmbito de comunicações multimídia. Em linhas gerais, essa grande variedade de protocolos com ênfase multimídia possibilita observar parte da complexidade que envolve (ou pode envolver) esse processo de comunicação multimídia. Como sugestão de leitura, os livros dos autores Kurose e Ross (2010) e Tanenbaum (2003), além das referências pertinentes às propostas dos protocolos descritos anteriormente, podem ser consultados para maiores informações sobre ambos.

### **2.1.3 – Distribuição de Conteúdos Multimídia na Internet**

Em linhas gerais, o processo de comunicação envolvendo aplicações multimídia ocorre no contexto Internet. Por sua vez, esse processo está sujeito ao serviço *best effort* dessa arquitetura, bem como às suas características quanto à temporização (atraso e variação de atraso), perda de pacotes e largura de banda, dentre outros. Dessa forma, parte das dificuldades voltadas ao acesso e distribuição de conteúdos multimídia na Internet se deve a essas características (enquanto requisitos das aplicações multimídia).

No âmbito do desenvolvimento, conforme descrito na Seção 2.1.1, algumas estratégias podem ser utilizadas para minimizar os efeitos desse serviço *best effort* às aplicações multimídia, tais como pela utilização de técnicas de compressão, protocolos específicos, *download* antecipado de conteúdos e envio de informações redundantes, dentre outras (KUROSE; ROSS, 2010).

No âmbito de Provedores e *Data Centers*, além de investimentos na infraestrutura de rede, tais como em enlaces, comutadores e servidores de alto desempenho, é comum a utilização de soluções de CDNs, *Cloud Computing* e de recursos e estratégias que simplifiquem e aproximem o conteúdo multimídia do cliente, tais como por meio de estruturas de *cache* e de Pontos de Troca de Tráfego (PTTs), dentre outros. Como exemplo, o trabalho de Adhikari et al. (2015) aponta a utilização de três CDNs (Akamai, Limelight e Level3) na distribuição de conteúdos de vídeo dos portais NetFlix e Hulu. Os trabalhos de Li,

X et al. (2017) e Cai et al. (2016) empregam abordagens no âmbito de *Cloud Computing*, tais como por meio de CaaS (*Caching as a Service*) e de *Cloud Gaming*, respectivamente, para a distribuição de conteúdos multimídia de vídeo e de jogos *online*, dentre outros, na rede. Por sua vez, abordagens sobre o comportamento da rede (Internet) em transmissões de grandes eventos esportivos, tais como o Super Bowl Americano de 2016 e a Copa das Confederações da FIFA de 2013, bem como sobre os clientes de vídeo ao vivo, também, em transmissões de grandes eventos esportivos, tal como a Copa do Mundo de Futebol da FIFA de 2014, e do ecossistema de PTTs no Brasil, ambas com contribuições no âmbito da distribuição de conteúdos multimídia na Internet, são discutidas, respectivamente, nos trabalhos de Morales e Bergstrom (2017), Almeida et al. (2015), Santos et al. (2016) e Brito et al. (2016).

Nesse cenário, outros fatores cruciais em termos de distribuição de conteúdos multimídia na Internet estão associados à QoS e QoE. Dada a heterogeneidade, complexidade e estrutura organizacional e funcional da rede, o fornecimento de níveis de qualidade que atendam os requisitos das aplicações e dos usuários é uma tarefa desafiadora, tanto para a indústria quanto para a academia, com inúmeras discussões e propostas sobre esses temas. Como exemplo, apenas no âmbito das SDNs, os trabalhos de Karakus e Durrezi (2017) e Guck et al. (2017), dentre outros, sintetizam uma série de contribuições voltadas ao fornecimento de recursos de QoS na rede. De modo similar, os trabalhos de Robitza et al. (2017) e Maia, Yehia e Errico (2015), por exemplo, também sintetizam uma série de outras contribuições, porém, nesse caso, voltadas à QoE percebida pelo usuário final, de modo geral.

Nesse contexto, é possível observar que a distribuição de conteúdos multimídia na Internet, em especial, com fluxos de tráfego do tipo *stream*, representa grande parte dos desafios no âmbito de comunicações multimídia. Em linhas gerais, esses desafios compreendem questões técnicas, organizacionais e até mesmo comerciais voltadas à rede e suas aplicações, bem como até a forma com a qual os conteúdos multimídia serão acessados e percebidos, em termos de QoE, por exemplo, pelos usuários finais ao longo da rede.

## **2.2 – Modelos de Predição**

No âmbito de comunicações multimídia, o emprego de modelos e recursos de predição pode ser utilizado, por exemplo, para que determinadas situações sejam previstas e/ou para que determinadas ações sejam realizadas de modo antecipado na rede.

Nesse cenário, alguns trabalhos recentes discorrem sobre o emprego de recursos de predição com ênfase em comunicações multimídia, tais como para o provisionamento e distribuição de conteúdos multimídia (KRYFTIS et al., 2016a; KRYFTIS et al., 2016b), para mecanismos de QoS e de QoE (COSTA FILHO et al., 2016; DUANMU et al., 2017) e para indexação, recuperação e análise de conteúdos multimídia (BARBOSA; MOURA; SANTOS, 2016; DIAS; WIVES; ROESLER, 2013), dentre outros. Além disso, no âmbito da rede, o emprego de recursos de predição também é importante para o planejamento e otimização de infraestruturas de rede, para mecanismos de alocação e de gerência de recursos de rede e para análises de fluxos de tráfego de rede, dentre outros (CENEDESE; TRAMARIN; VITTURI, 2017; MIJUMBI et al., 2017; MORALES et al., 2017; SILVA-JUNIOR; KAMIENSKI, LORENA, 2014).

Diante desse cenário, diversos métodos e modelos de predição podem ser empregados, tais como por meio do filtro de Kalman, de redes neurais ou de lógica Fuzzy, dentre outros. Dentre tais métodos e modelos, o filtro de Kalman foi o escolhido para o fornecimento dos recursos de predição empregados neste trabalho. Essa escolha foi embasada, em especial, pela ampla aceitação e utilização do filtro de Kalman para previsões em tempo real, tais como para previsões de fluxos de tráfego de rede, bem como por sua utilização em outros projetos/trabalhos já realizados pelo grupo de trabalho ao qual esta Tese está vinculada (GOMNI, FT/UNICAMP).

Nesse contexto, as Seções seguintes discorrem sobre o filtro de Kalman e o subfiltro de Kalman.

### **2.2.1 – Filtro de Kalman**

De acordo com Welch e Bishop (2001), o filtro de Kalman (KALMAN, 1960) é, essencialmente, um conjunto de equações matemáticas que possibilitam a implementação de um estimador do tipo preditor-corretor. Em linhas gerais, essas equações podem ser algebricamente manipuladas de diversas formas. Uma dessas formas, descrita no trabalho de Ursini et al. (2014), é composta pelas seguintes equações (de observação e de modelagem):

$$Y_k = HX_k + V_k, \quad (1)$$

$$X_{k+1} = \Phi X_k + LU_k + \Gamma W_k. \quad (2)$$

Em que:

- $Y_k$  são as saídas do filtro (valores observados);
- $H$  é a matriz de observação;
- $X_k$  é o vetor de variáveis de estado =  $[x_k \dot{x}_k]^T$ ;
- $\Phi$  é o modelo de transição de estado que é aplicado ao estado anterior  $X_{k-1}$ ;
- $L$  é a matriz de ganho;
- $V_k$  e  $W_k$  são vetores de ruído de média zero ( $V_k \in R^m$  e  $W_k \in R^n$ ), em que  $R$  é um conjunto de números reais;
- $\Gamma$  é matriz de ruído de modelagem;
- $U_k$  representa as entradas de controle (para predição,  $U_k = 0$ ).

Nesse cenário, a covariância do vetor de ruído de modelagem é dada por  $Q_k = E[W_k W_k^T]$  e a covariância do vetor de ruído de observação é dada por  $R_k = E[V_k V_k^T]$ .

Diante desse cenário, conforme descrito por Welch e Bishop (2001), no algoritmo do filtro de Kalman, um processo é estimado usando uma forma de controle por realimentação: o filtro estima o estado do processo em dado instante e obtém, então, a realimentação na forma de medidas ruidosas. Dessa forma, as equações do filtro de Kalman podem ser representadas por dois grupos: equações de atualização de tempo (equações de predição) e equações de atualização de medição (equações de correção). Segundo os autores, as equações de atualização de tempo são as responsáveis por projetar com antecedência (no tempo) as estimativas da covariância do estado atual e do erro, para obter as estimativas *a priori* para a próxima etapa. Por sua vez, as equações de atualização de medição são responsáveis pela realimentação, ou seja, para incorporar uma nova medição na estimativa *a priori* para obter, posteriormente, uma estimativa melhorada. Em linhas gerais, essas equações também podem ser algebricamente manipuladas de diversas formas, tais como segue:

- Equações de atualização de tempo (equações de predição):

$$X_k^* = \Phi \bar{X}_k, \quad (3)$$

$$r_k = Y_k - H X_k^*, \quad (4)$$

$$P_{k+1}^* = \Phi \bar{P}_k \Phi^r + Q_k. \quad (5)$$

- Equações de atualização de medição (equações de correção):

$$K_k = P_k^* H^T (\sigma_k^2)^{-1}, \quad (6)$$

$$\bar{X}_k = X_k^* + K_k r_k, \quad (7)$$

$$\bar{P}_k = (I - K_k H) P_k^*. \quad (8)$$

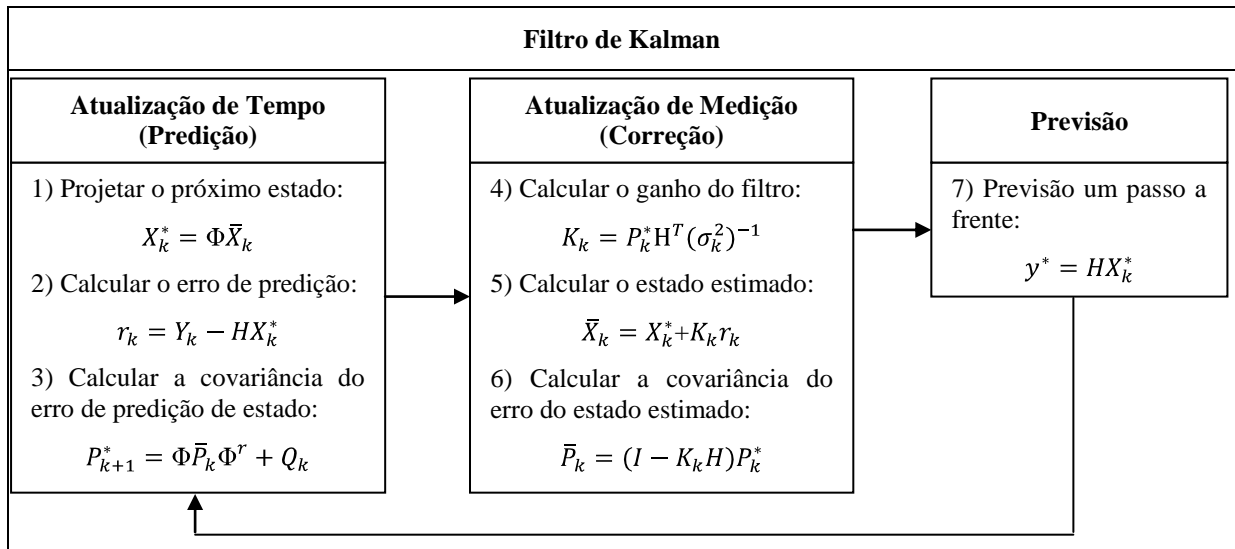
Em que:

- $\Phi$ ,  $X_k$ ,  $Y_k$  e  $H$  são pertinentes as Equações 1 e 2;
- $X_k^*$  representa a predição do estado  $X_k$  a partir do estado estimado do instante anterior  $\bar{X}_k$ , e  $\bar{X}_k$  a atualização dessa estimativa com base na leitura efetuada no instante  $k$  (filtragem);
- $r_k$  representa o erro de predição entre a saída  $Y_k$  e sua previsão  $y^* = HX_k^*$ ;
- $P_k^*$  é a matriz de covariância do erro de predição;
- $\bar{P}_k$  é a covariância do erro do estado estimado;
- $K_k$  é a matriz de ganho do filtro (ganho de Kalman);
- $\sigma_k^2$  é a covariância de estado do filtro.

Nesse cenário, quanto às equações de atualização de tempo (equações de predição), a Equação 3 é a responsável por projetar o próximo estado, a Equação 4 é a responsável por calcular o erro de predição e, por sua vez, a Equação 5 é a responsável por calcular a covariância do erro de predição de estado. Por sua vez, quanto às equações de atualização de medição (equações de correção), a Equação 6 é a responsável por calcular o ganho do filtro (ganho de Kalman), a Equação 7 é a responsável por calcular o estado estimado (valor filtrado) e, por sua vez, a Equação 8 é a responsável por calcular a covariância do erro do estado estimado.

Nesse contexto, a Figura 2 ilustra os passos realizados pelo algoritmo discreto do filtro de Kalman quanto a tais equações, de modo geral.

**Figura 2.** Passos realizados pelo algoritmo discreto do filtro Kalman. Fonte: Adaptado de Oliveira (2011).



É importante destacar que esse processo de cálculo é realizado de modo recursivo, sendo esta uma das principais características e atrativos do filtro de Kalman (WELCH; BISHOP, 2001).

### 2.2.2 – Subfiltro de Kalman

De acordo com Ursini et al. (2014), para que o método de Kalman convirja, ele necessita que a relação de covariâncias  $Q_k/R_k$  mantenha-se estável, dentro de uma margem de tolerância, durante o procedimento de cálculo.

Nesse contexto, conforme descrito no trabalho de Welch e Bishop (2001), na implementação do filtro de Kalman, a covariância do ruído  $R$  ( $R_k$  neste trabalho) é geralmente medida antes da operação do filtro (utilizando algumas medições *off-line*). Por outro lado, a determinação da covariância do ruído  $Q$  ( $Q_k$  neste trabalho) é geralmente mais trabalhosa, uma vez que, normalmente, não temos a capacidade de observar diretamente o processo que estamos estimando. Nesse cenário, de acordo com os autores, o melhor desempenho do filtro (estatisticamente falando) pode ser obtido ajustando os parâmetros  $Q$  e  $R$ . Para tal, esse ajuste é geralmente realizado de modo *off-line*, com a utilização de outro filtro.

Nesse cenário, este trabalho emprega um filtro adicional ao filtro de Kalman, proposto por Rios Neto e Kuga (1982) e tido como subfiltro de Kalman, para realizar esse ajuste de parâmetros. De acordo com Ursini et al. (2014), a solução proposta por esse método é efetuar continuamente o recálculo dos valores das matrizes  $Q_k$  e  $R_k$  a cada passo de interação, de modo a manter a relação  $Q_k/R_k$  estável no entorno do respectivo estado. Para tal, as matrizes  $Q_k$  e  $R_k$  também são recalculadas. Esse recálculo é realizado pelo subfiltro por meio das seguintes equações:

- Equações de atualização de tempo (equações de predição):

$$q_{k+1}^* = \Phi^q \bar{q}_k, \quad (9)$$

$$P_{k+1}^{q*} = \Phi^q \bar{P}_k^q (\Phi^q)^T + \sigma_q^2. \quad (10)$$

- Equações de atualização de medição (equações de correção):

$$K_{k+1}^q = P_{k+1}^{q*} \cdot M^T (MP_{k+1}^{q*} M^T + E[\eta_{k+1}^2])^{-1}, \quad (11)$$

$$\bar{q}_{k+1} = q_{k+1}^* + K_{k+1}^q (Z_{k+1} - Mq_{k+1}^*), \quad (12)$$

$$\bar{P}_{k+1}^q = P_{k+1}^{q*} - K_{k+1}^q MP_{k+1}^{q*}. \quad (13)$$

Nesse cenário, quanto às equações de atualização de tempo (equações de predição), as Equações 9 e 10 correspondem ao valor do estado predito e à covariância do estado predito, respectivamente. Por sua vez, quanto às equações de atualização de medição (equações de correção), a Equação 11 é a responsável pelo cálculo do ganho do subfiltro, a Equação 12 é a responsável pelo cálculo do valor estimado do estado do subfiltro e, por sua vez, a Equação 13 é a responsável pelo cálculo da covariância do estado estimado do subfiltro.

De modo complementar a tais equações, as equações de pseudo-observação e de modelagem de estado para o subsistema são dadas, respectivamente, pelas Equações 14 e 15 como seguem:

$$Z_{k+1} = Mq_{k+1} + \eta_{k+1}, \quad (14)$$

$$q_{k+1} = \Phi^q q_k + W_k^q. \quad (15)$$

A equação da covariância de pseudo-observação é dada pela Equação 16.

$$E[\eta_{k+1}^2] = 4r_{k+1}^2 + 2R_{k+1}^2. \quad (16)$$

Em que  $\sigma_q^2$  representa a covariância de estado do subfiltro,  $q_0^* = [1 \ 1]^T$  e  $\sigma_q^2 = 10^{-32} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ . Nesse cenário,  $Q_k$  é atualizado da seguinte maneira:

$$j = 1,2 \begin{cases} \text{if } q_{k+1}^*(j, 1) \geq 0 \rightarrow Q(j, j) = q_{k+1}^*(j, 1) \\ \text{else } Q_{jj} = 0. \end{cases}$$

Por sua vez, os valores iniciais são dados como segue:

$$Q_k = \begin{bmatrix} \sigma_{1k}^2 & 0 \\ 0 & \sigma_{2k}^2 \end{bmatrix},$$

$$\Phi = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; X_k = [X_k \dot{X}_k]; X_0 = Y_1; e \dot{X}_0 = 0,$$

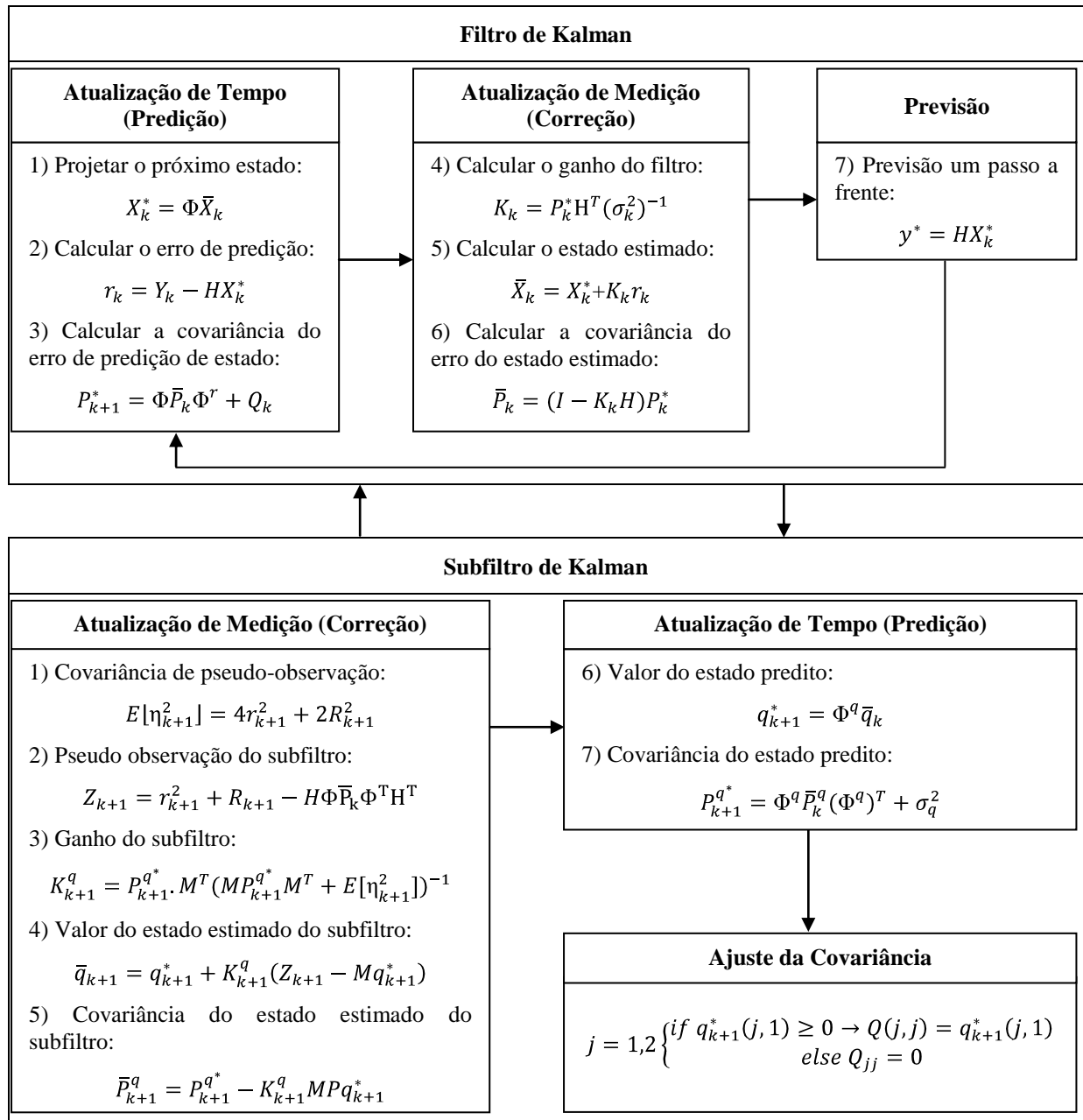
$$P_0^* = \bar{P}_0; \Phi^q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; M = [1 \ 0].$$

Nesse cenário, conforme descrito no trabalho de Ursini et al. (2014), após a execução do subfiltro de Kalman, o método executa o algoritmo tradicional do filtro de Kalman. Nesse caso, em vez de se delimitar um espaço de valores estático dentro do qual a relação  $Q_k/R_k$  (inicialmente calculada) é válida, o método utilizando o subfiltro efetua uma reavaliação contínua de  $Q_k$  e  $R_k$ , realizando eventuais ajustes nos valores que compõem as matrizes. Dessa forma, é possível manter a equivalência matemática da relação entre  $Q_k$  e  $R_k$ , de modo que, a cada novo passo do algoritmo, essa relação seja sempre válida dentro de limites pré-estabelecidos.

Nesse contexto, a Figura 3 ilustra os passos realizados pelo algoritmo discreto do filtro de Kalman e pelo subfiltro de Kalman quanto a tais equações, de modo geral.



**Figura 3.** Passos realizados pelo algoritmo discreto do filtro Kalman e pelo subfiltro de Kalman. Fonte: Adaptado de Oliveira (2011).



### 2.3 – Redes Definidas por *Software*

Com uma abordagem centrada na separação dos planos de dados (encaminhamento) e de controle, as Redes Definidas por *Software* representam um paradigma emergente que objetiva simplificar e flexibilizar o gerenciamento da rede, além de permitir e facilitar sua evolução e inovação (KREUTZ et al., 2015; XIA et al., 2015; NUNES et al.,

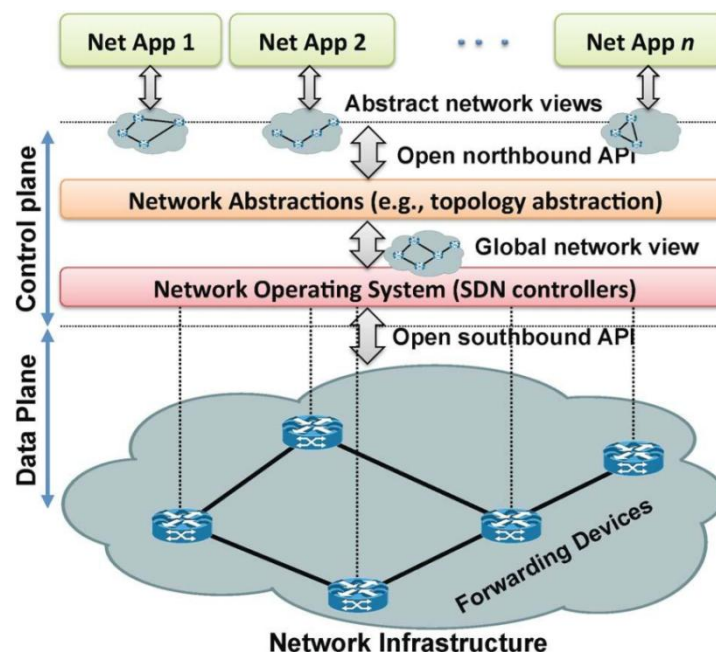
2014). Além disso, as SDNs também fornecem recursos que contribuem com algumas das limitações associadas às infraestruturas de redes atuais, tais como limitações de gerenciamento, de escalabilidade e de segurança, dentre outras (MENDIOLA et al., 2017; NGUYEN et al., 2017; RAWAT; REDDY, 2017).

Nesse contexto, os tópicos seguintes discorrem sobre conceitos e fundamentos voltados às SDNs, com ênfase quanto sua arquitetura, principais componentes e princípios básicos e quanto ao protocolo OpenFlow.

### 2.3.1 – Arquitetura, Principais Componentes e Princípios Básicos

Conforme descrito nos trabalhos de Kreutz et al. (2015), Xia et al. (2015) e Nunes et al. (2014), em SDN, que possui infraestrutura física similar às redes tradicionais e cuja arquitetura e abstrações fundamentais são ilustradas na Figura 4, a inteligência da rede está centralizada em controladores baseados em *software* (plano de controle), ou seja, no controlador SDN (ou NOS, *Network Operating System*), com dispositivos simples, e programáveis por meio de uma interface aberta, realizando o encaminhamento de pacotes (plano de dados).

**Figura 4.** Arquitetura SDN e suas abstrações fundamentais. Fonte: Kreutz et al. (2015).



Nesse cenário, o controlador SDN é o elemento de *software* capaz de consolidar os diversos dispositivos do plano de dados, abstraindo as particularidades desses e fornecendo uma visão global sobre a rede (KREUTZ et al., 2015; XIA et al., 2015; NUNES et al., 2014). Por meio do controlador, esses dispositivos do plano de dados também podem ser programados e gerenciados de maneira centralizada e abstrata, sem a necessidade de configurações individuais ou o domínio de seus eventuais distintos *softwares* de controle. Dessa forma, os dispositivos do plano de dados podem tratar apenas de questões envolvendo o encaminhamento de pacotes. Num contexto mais amplo, os fabricantes desses dispositivos podem concentrar seus esforços apenas no nível de *hardware* de encaminhamento, não necessitando desenvolver os *softwares* de controle aos mesmos, mas apenas uma interface para que o controlador SDN possa utilizá-los e gerenciá-los. De acordo com Nunes et al. (2014), essa estrutura, além de possibilitar a consolidação dos diversos dispositivos da rede, facilita a implantação de novos protocolos e aplicações, bem como na forma como a rede é vista e gerenciada, com amplo controle sobre seus fluxos de tráfego.

Diante desse cenário, de modo a objetivar a compatibilidade e interoperabilidade entre os diversos dispositivos dos planos de dados e o controlador, toda a comunicação entre ambos se dá por meio de interfaces abertas e padronizadas de comunicação, algo difícil em redes tradicionais, em especial, devido à grande variedade de interfaces proprietárias (e fechadas) e pela natureza distribuída do plano de controle (KREUTZ et al., 2015).

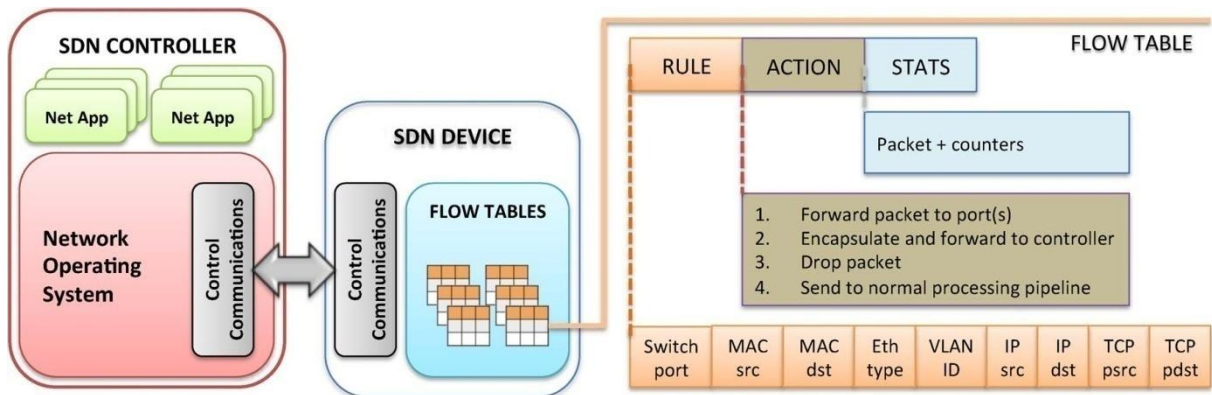
Nesse contexto, conforme ilustrado na Figura 4, toda comunicação entre o controlador e os dispositivos do plano de dados ocorre por meio da interface *southbound*, também chamada de API *southbound* ou SI (*southbound interface*). Nesse cenário, no âmbito da interface *southbound*, o protocolo OpenFlow (MCKEOWN et al., 2008) tem sido amplamente utilizado para permitir a comunicação entre os dispositivos do plano de dados e o controlador, com inúmeras contribuições em SDN associadas ao mesmo (LARA; KOLASANI; RAMAMURTHY, 2014; HU; HAO; BAO, 2014). Por sua vez, por meio da interface *northbound*, também chamada de API *northbound* (ou NI, *northbound interface*), aplicações de rede podem interagir com o controlador SDN e, por meio desse, com a rede, de modo geral. Nesse cenário, pelo fato de não existir a definição de um padrão para a API *northbound*, a conexão entre as aplicações e o controlador pode ocorrer de diversos modos, tais como por meio de uma API REST, de uma API específica disponibilizada pelo controlador ou por meio de linguagens específicas de programação (FOSTER et al., 2013), dentre outras.

De acordo com Kreutz et al. (2015) e Nunes et al. (2014), essa abordagem baseada na utilização de interfaces abertas e padronizadas de comunicação é uma das principais responsáveis pelos recursos de inovação e evolução proporcionados pelas SDNs. Segundo os autores, essa abordagem facilita, por exemplo, a implantação de novos protocolos e aplicações e não se restringe à utilização de interfaces fechadas e limitadas dos fabricantes dos dispositivos de *hardware*, algo tido como difícil em redes tradicionais, em especial, devido à grande variedade de dispositivos no plano de dados (com interfaces proprietárias e fechadas), bem como da natureza distribuída do plano de controle.

Quanto ao encaminhamento de pacotes, de acordo com Kreutz et al. (2015), na arquitetura SDN as decisões de encaminhamento são baseadas em fluxos, e não apenas no endereço do destinatário. De acordo com os autores, no contexto SDN/OpenFlow, um fluxo é uma sequência de pacotes entre uma origem e um destino, os quais recebem políticas de serviços idênticas nos dispositivos de encaminhamento da rede. Dessa forma, os autores argumentam que a abstração por fluxos permite unificar o comportamento de diferentes tipos de dispositivos de rede, tais como roteadores, *switches*, *firewalls* e *middleboxes*, além de possibilitar uma flexibilidade de programação sem precedentes quanto a ambos (limitada, apenas, às capacidades das tabelas de fluxo implementadas em tais dispositivos). Em linhas gerais, essa abordagem possibilita múltiplas opções de gerência e de controle quanto aos fluxos de tráfego de rede, tais como no âmbito da engenharia de tráfego (MENDIOLA et al, 2017), da QoS (KARAKUS; DURRESI, 2017) e da segurança da rede (RAWAT; REDDY, 2017), dentre outras.

Diante desse cenário, os dispositivos de encaminhamento do plano de dados da SDN tratam do encaminhamento de pacotes com base em suas tabelas de fluxos, em que cada dispositivo pode ter uma ou mais tabelas de fluxos. Por sua vez, tal como descrito no trabalho de Kreutz et al. (2015), no âmbito do protocolo OpenFlow, cada regra de fluxo pode ser configurada combinando múltiplos *matching fields*, com ações específicas de encaminhamento ou de descarte, por exemplo, além da possibilidade de se obter estatísticas sobre os pacotes processados. Como exemplo, a Figura 5 ilustra essa organização em termos das tabelas de fluxos em dispositivos OpenFlow, bem como da possibilidade de combinação de múltiplos *matching fields* na definição das regras de fluxos, das ações a serem executadas sobre os fluxos correspondentes e da possibilidade de se obter estatísticas quanto aos mesmos.

**Figura 5.** Decisões de encaminhamento baseadas em fluxos em dispositivos SDN OpenFlow.  
 Fonte: Kreutz et al. (2015).



Nesse cenário, no dispositivo OpenFlow, os pacotes são tratados com base em suas correspondências em termos de fluxo. De acordo com Kreutz et al. (2015), assim que recebe um pacote, o dispositivo de encaminhamento procura por uma correspondência em suas tabelas de fluxo. Essa busca inicia na primeira tabela de fluxos do dispositivo e finaliza assim que uma correspondência seja localizada ou na inexistência da mesma (quando o pacote não se enquadra em nenhuma regra de fluxo presente no dispositivo). Nesse caso, o pacote pode ser descartado, enviado ao controlador para que uma ação seja executada ou processado respeitando uma regra padrão presente no dispositivo. A prioridade das regras segue o número de sequência natural das tabelas de fluxo e, em cada tabela, respeita sua ordem em termos de linhas de processamento. No que se refere às ações que serão executadas aos fluxos correspondentes, além daquelas ilustradas na Figura 5, Kreutz et al. (2015) também discorrem da possibilidade de enviar o pacote para uma próxima tabela de fluxo ou para tabelas de fluxos especiais, tais como para tabelas de grupos ou de medição para mecanismos de QoS, dentre outros.

Dessa forma, de acordo com Kreutz et al. (2015), os princípios que fundamentam e servem de pilares para a arquitetura SDN podem ser resumidos:

1. Na separação dos planos de dados e de controle;
2. Na centralização da lógica de controle (“inteligência” da rede) em uma entidade externa aos dispositivos do plano de dados (nesse caso, no controlador SDN ou NOS);

3. Na possibilidade de programar a rede por meio de aplicações rodando no topo dos NOSs que, por sua vez, é o responsável por interagir com os dispositivos do plano de dados;
4. Em decisões de encaminhamento baseadas em fluxos.

Por sua vez, os principais componentes de sua arquitetura, podem ser resumidos:

1. No controlador SDN ou NOS;
2. Nos dispositivos de encaminhamento do plano de dados, habilitados para SDN;
3. Nos protocolos e interfaces que viabilizam a arquitetura SDN.

Nesse contexto, considerando os princípios básicos quanto sua arquitetura e estrutura funcional, bem como dos componentes empregados na aplicação de seu paradigma, as Redes Definidas por *Software*, além de possibilitarem o desenvolvimento de pesquisas inovadoras sobre as atuais redes IP (KREUTZ et al., 2015; XIA et al., 2015; NUNES et al., 2014; LARA; KOLASANI; RAMAMURTHY, 2014; GUEDES et al., 2012, MCKEOWN et al., 2008), também surgem como potencial arquitetura para a “Internet do Futuro” (HAKIRI et al., 2014; GUEDES et al., 2012), concentrando esforços da academia e, também, da indústria em seu estudo e disseminação (MENDIOLA et al, 2017; KARAKUS; DURRESI, 2017; RAWAT; REDDY, 2017; YAP et al.; 2017; JAIN et al., 2013).

### **2.3.2 – Protocolo OpenFlow**

No âmbito da comunicação entre o controlador e os dispositivos do plano de dados, o OpenFlow (MCKEOWN et al., 2008) é o protocolo mais utilizado para a implantação da arquitetura SDN (KREUTZ et al., 2015; XIA et al., 2015; LARA; KOLASANI; RAMAMURTHY, 2014; HU; HAO; BAO, 2014). Alguns autores, tais como Xia et al. (2015), por exemplo, discorrem que a implantação SDN “*de facto*” se dá por meio do protocolo OpenFlow.

Nesse contexto, de acordo com Lara, Kolasani e Ramamurthy (2014), enquanto a arquitetura SDN fornece uma abstração sobre a rede, o protocolo OpenFlow faz o mesmo em

termos dos dispositivos do plano de dados, sendo o responsável por abstrair suas particularidades individuais, além de padronizar todas as comunicações entre o controlador e tais dispositivos. Além disso, pelo fato de poder ser empregado, por exemplo, em dispositivos de *hardware* tradicionais, em *software switches* e em uma série de simuladores e emuladores de rede, tornou-se amplamente aceito pela academia e, também, pela indústria, com soluções comerciais disponibilizadas por grandes empresas, além de implementações em grandes Provedores e *Data Centers*, tais como no Google<sup>8</sup> (YAP et al.; 2017; JAIN et al., 2013) e em redes experimentais como OFELIA<sup>9</sup> e GENI<sup>10</sup>, dentre outras.

No que se refere às suas versões, conforme descrito em Xia et al. (2015), as especificações sobre o OpenFlow estão em constante evolução, com novos recursos sendo lançados a cada nova versão. Suas primeiras versões foram lançadas pelo *OpenFlow Switch Consortium*<sup>11</sup> e, posteriormente, transferidas para a *Open Networking Foundation (ONF)*<sup>12</sup>, com intuítos de padronização, divulgação e adoção de SDNs utilizando padrões abertos. Caso o leitor se interesse, suas especificações mais recentes podem ser consultadas por meio da seguinte URL: <https://www.opennetworking.org/software-defined-standards/specifications/>.

De modo complementar ao descrito por Xia et al. (2015), de acordo com Kreutz et al. (2015), em cada nova versão do OpenFlow novos campos de ação, tais como campos Ethernet, IPv4/IPv6, MPLS e TCP/UDP, por exemplo, também são disponibilizados para a definição de regras de fluxos, porém, apenas alguns desses campos são obrigatórios para estar em conformidade com uma dada versão do protocolo. Da mesma forma, algumas ações e tipos de portas também são opcionais. Ainda assim, as regras de fluxos podem ser definidas por meio da combinação de múltiplos *matching fields*, tal como ilustrado, anteriormente, na Figura 5, o que fornece ampla flexibilidade quanto suas definições.

Em termos arquiteturais, de acordo com Hu, Hao e Bao (2014), o protocolo OpenFlow possui 3 componentes típicos: (1) *switches* OpenFlow, (2) controladores OpenFlow e (3) tabelas de fluxos. Nesse cenário, toda comunicação entre o controlador e os *switches* se dá por meio do protocolo OpenFlow, utilizando um canal seguro entre ambos. Cabe ao controlador gerenciar os *switches* e suas tabelas de fluxos e aos *switches*, com base em suas tabelas de fluxos, tratar de questões envolvendo o encaminhamento de fluxos ao

<sup>8</sup> Google em <https://www.google.com.br>

<sup>9</sup> OFELIA em <http://www.fp7-ofelia.eu>

<sup>10</sup> GENI em <http://www.geni.net>

<sup>11</sup> *OpenFlow Switch Consortium* em <http://archive.openflow.org>

<sup>12</sup> *Open Networking Foundation (ONF)* em <https://www.opennetworking.org/>

longo da rede. Por sua vez, o processo de encaminhamento de pacotes ao longo da rede é baseado em fluxos, tal como descrito na Seção anterior.

Nesse cenário, como exemplo de dispositivos de encaminhamento habilitados para OpenFlow, grandes empresas como Cisco<sup>13</sup>, HP<sup>14</sup> e Juniper<sup>15</sup>, dentre outras, possuem *switches* habilitados ao mesmo. De modo complementar, soluções de *software switches*, tais como Open vSwitch<sup>16</sup> e OpenFlow 1.3 Software Switch<sup>17</sup>, por exemplo, também são comumente empregadas na experimentação e prototipação de SDNs OpenFlow. No nível de controladores, há uma grande variedade de opções, tais como Floodlight<sup>18</sup>, OpenDaylight<sup>19</sup> e NOX<sup>20</sup>, dentre outros, com grande parte desses suportando apenas o OpenFlow enquanto API *southbound* (KREUTZ et al., 2015). Há ainda a possibilidade de sua utilização em emuladores e simuladores, tais como Mininet<sup>21</sup>, Mininet-WiFi<sup>22</sup> e ns-3<sup>23</sup>, dentre outros.

De modo complementar, tal como descrito por Kreutz et al. (2015), cabe ressaltar que, além do OpenFlow, o controlador SDN pode utilizar diferentes APIs *southbound* simultaneamente, tais como ForCES (HALEPLIDIS et al., 2015) e *Protocol Oblivious Forwarding (POF)* (LI, S et al., 2017), dentre outras. Dessa forma, de acordo com os autores, essa possibilidade de utilização de várias APIs, além de *plugins* para protocolos de monitoramento e gerência, por exemplo, enquanto interface *southbound*, é essencial em termos de compatibilidade e heterogeneidade da rede, uma vez que o plano de dados pode conter uma grande quantidade de dispositivos físicos e, também, virtuais, com eventual variedade de interfaces disponibilizadas por ambos.

## 2.4 – Trabalhos Relacionados

Trabalhos recentes discorrem sobre questões envolvendo comunicações multimídia em ambientes SDN OpenFlow. As discussões são amplas, compreendendo

---

<sup>13</sup> Soluções SDN Cisco em <https://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html>

<sup>14</sup> Soluções SDN HP em <https://www.hpe.com/us/en/networking/infrastructure.html>

<sup>15</sup> Soluções SDN Juniper em <https://www.juniper.net/us/en/products-services/sdn/>

<sup>16</sup> Open vSwitch em <http://openvswitch.org/>

<sup>17</sup> OpenFlow 1.3 Software Switch em <https://github.com/CPqD/ofsoftswitch13>

<sup>18</sup> Floodlight em <http://www.projectfloodlight.org/floodlight>

<sup>19</sup> OpenDaylight em <https://www.opendaylight.org>

<sup>20</sup> NOX em <http://www.noxrepo.org>

<sup>21</sup> Mininet em <http://mininet.org>

<sup>22</sup> Mininet-WiFi em <https://github.com/intrig-unicamp/mininet-wifi>

<sup>23</sup> Ns-3 em <https://www.nsnam.org>



abordagens no âmbito do encaminhamento/roteamento e da QoS, tais como por meio de mecanismos de roteamento dinâmico/adaptativo e de classificação e priorização para fluxos de tráfego multimídia, bem como no âmbito de recursos de cache e de predição, dentre outras. Nesse contexto, esta Seção discorre sobre alguns dos trabalhos relacionados com esta Tese, em especial, quanto aos objetivos gerais e específicos delineados na Seção 1.2.

Como exemplo, no âmbito do encaminhamento/roteamento e da QoS para fluxos de tráfego multimídia, os trabalhos de Owens e Durresi (2017), Adami et al. (2015a), Adami et al. (2015b) e Egilmez e Tekalp (2014), dentre outros, propõem o emprego de mecanismos de roteamento dinâmico no projeto do controlador OpenFlow para o fornecimento de QoS aos fluxos multimídia durante seu processo de encaminhamento ao longo da rede. Em linhas gerais, ambas as propostas consistem em rotear os fluxos multimídia de modo distinto dos demais fluxos da rede, porém com abordagens diferenciadas quanto às restrições consideradas no processo de roteamento, bem como nos níveis de classificação ofertados aos fluxos multimídia. De modo complementar, no trabalho de Adami et al. (2015a) os autores também consideram o emprego de um módulo para inspeção de fluxos de tráfego VoIP (via SIP) e de classes de tráfego nos *switches* do plano de dados da SDN.

Ainda no âmbito da QoS, abordagens diferenciadas são exploradas para possibilitar a identificação e classificação dos fluxos de tráfego da rede, bem como na forma com a qual a QoS será efetivamente empregada no ambiente SDN. Como exemplo, no trabalho de Amelyanovich et al. (2017), os autores exploram a utilização de rótulos DSCP para identificar e classificar os diferentes fluxos de tráfego de rede no âmbito de *Wireless LANs* baseadas em SDN. A proposta também considera o emprego de filas com prioridades distintas nos *switches* da SDN para a oferta de diferentes níveis de QoS aos fluxos de tráfego Wi-Fi.

De modo similar, outras abordagens que também consideram a utilização de rótulos DSCP para a diferenciação de fluxos de tráfego de rede são descritas nos trabalhos de Silva et al. (2013) e López-Rodríguez e Campelo (2014). Como exemplo, em um contexto interdomínios, no trabalho de Silva et al. (2013), os autores propõem uma arquitetura baseada em virtualização de redes para o fornecimento de QoS fim-a-fim em redes OpenFlow. A proposta considera dois níveis de mapeamento: o primeiro nível mapeia especificações de QoS (QSPEC) entre fluxos OpenFlow e o esquema de prioridades do PCP (*Priority Code Point*) e o segundo nível fornece mapeamento e interoperabilidade interdomínios por meio do protocolo NSIS (*Next Steps in Signaling*). Nesse caso, o mapeamento L2/L3 é realizado por

meio da conversão do campo PCP para o campo DSCP, o qual é realizado por *gateways* com suporte ao OpenFlow e ao NSIS. Por sua vez, no âmbito de *Service Providers*, uma arquitetura baseada em SDN é apresentada por López-Rodríguez e Campelo (2014). Nessa proposta, o protocolo MPLS (*Multiprotocol Label Switching*) é empregado no plano de dados e o fornecimento de recursos de QoS se dá por meio de regras e tabelas específicas para tal, com diferenciação entre fluxos de tráfego sendo realizada por diferentes *matching fields*, tais como por meio do DSCP e o do MPLS EXP, dentre outros.

De modo complementar, alguns trabalhos recentes também exploram resoluções DNS e informações dos protocolos de aplicação e de portas de transmissão para identificar e classificar os diferentes fluxos de tráfego da rede, bem como do emprego de múltiplos escalonadores de pacotes no âmbito da QoS. Como exemplo, no âmbito de *home networks* em SDN, nos trabalhos de Seddiki et al. (2015) e Seddiki et al. (2014), as classificações de fluxo se dão por meio de resoluções DNS e pela identificação do protocolo de aplicação, com QoS fornecida, em especial, por meio de recursos de *traffic shaping* associados à grupos de aplicações que terão sua taxa de transmissão limitada na rede local. No trabalho de Karaman et al. (2015), os autores classificam e priorizam fluxos multimídia de áudio e vídeo, em transmissões VoIP embasadas no protocolo SIP, por meio de interações entre o servidor SIP e o controlador SDN (via RESTful API), nesse caso, com base nas portas de transmissão (de origem e de destino) empregadas nas sessões de áudio e vídeo. A proposta também considera o emprego de filas com prioridades e de recursos de limitação de banda nos *switches* da SDN para o fornecimento de recursos de QoS na rede. Por sua vez, no trabalho de Ishimori et al. (2013), os autores propõem a utilização de múltiplos escalonadores de pacotes para melhorar o fornecimento de recursos de QoS em redes OpenFlow. A ideia é estender o controle sobre escalonadores de pacotes do Linux, como HTB (*Hierarchical Token Bucket*), RED (*Randomly Early Detection*) e SFQ (*Stochastic Fairness Queuing*) para ambientes OpenFlow. A proposta também considera recursos de *enqueueing* e *traffic shaping* para alcançar QoS e QoE.

Quanto ao emprego de recursos de cache no âmbito multimídia em SDN, os trabalhos de Chiang e Li (2016), Trajano e Fernandez (2016) e Georgopoulos et al. (2015), por exemplo, discorrem sobre abordagens nesse tema. Em linhas gerais, ambos os trabalhos exploram recursos de cache para aprimorar a distribuição de conteúdos multimídia na rede, porém com abordagens diferenciadas quanto à arquitetura e os componentes propostos, bem como quanto ao conteúdo multimídia objetivo do cache, dentre outros. Como exemplo, os trabalhos de Chiang e Li (2016) e Georgopoulos et al. (2015) discorrem sobre arquiteturas de

cache com ênfase na distribuição de conteúdos de vídeo sob demanda (*Video-on-Demand, VoD*) em ambientes SDN, ambos com arquiteturas e componentes específicos em termos de *cache controllers* e *cache nodes*, dentre outros. De modo complementar ao trabalho de Georgopoulos et al. (2015), o trabalho de Broadbent et al. (2015) apresenta e explora as APIs utilizadas para interagir com a solução proposta pelos autores (nomeada de OpenCache). Por sua vez, no trabalho de Trajano e Fernandez (2016), os autores propõem uma arquitetura de cache mais flexível quanto aos conteúdos multimídia suportados, ambos experimentados sobre HTTP e em uma arquitetura composta por dois componentes principais, tidos como ContentSDN Proxy e ContentSDN Cache, responsáveis, respectivamente, por lidar com as requisições realizadas pelos clientes da rede e pelo armazenamento/entrega do conteúdo em cache, de modo geral.

Por sua vez, nos trabalhos de Kryftis et al. (2016a), Kryftis et al. (2016b) e Costa Filho et al. (2016), por exemplo, os autores discorrem sobre o emprego de recursos de predição no âmbito de comunicações multimídia. Nesse caso, nos trabalhos de Kryftis et al. (2016a) e Kryftis et al. (2016b), os autores propõem uma arquitetura de rede, que explora recursos de predição por meio de uma RPE (*Resource Prediction Engine*), para o provisionamento e entrega de conteúdos multimídia na rede. Em linhas gerais, por meio de informações de monitoramento e de dados históricos sobre demandas por conteúdos multimídia realizados no passado, a arquitetura proposta utiliza recursos de predição para prever demandas futuras sobre tais conteúdos. As informações obtidas são utilizadas, em especial, para alocações de recursos de banda e para a seleção dos métodos de distribuição do conteúdo multimídia ao longo da rede. Por sua vez, no trabalho de Costa Filho et al. (2016), os autores propõem um modelo de predição de qualidade de *streaming* de vídeo com base na observação de desempenho da rede IP subjacente. Para tal, o modelo proposto considera a utilização de alguns indicadores de rede/QoS (como atraso e variação do atraso, por exemplo) e de aplicação (como tempo de inicialização e número de paradas, por exemplo) em sua abordagem. A partir de tais indicadores, o modelo proposto explora a técnica de árvores de decisão para estabelecer e quantificar a relação entre indicadores de QoS e indicadores objetivos que infiram a qualidade da reprodução de vídeos no contexto do usuário final. De modo complementar, questões envolvendo a predição de fluxos de rede são discutidas no trabalho de Silva-Junior, Kamienski e Lorena (2014). Em sua abordagem, os autores utilizam algoritmos de análise de redes complexas e técnicas de aprendizado de máquina para prever os fluxos que se manterão e que existirão em um instante futuro no âmbito de SDNs. Para tal,

os autores utilizam alguns conjuntos de dados de duas redes de aplicação (P2P e e-mail), os quais são mapeados em modelos topológicos e utilizados como redes físicas subjacentes.

Nesse contexto, diante dos trabalhos relacionados, a abordagem proposta nesta Tese não considera modificações no projeto do controlador OpenFlow ou nos *switches* do plano de dados da SDN para o fornecimento de QoS na rede, tal como nos trabalhos de Owens e Duresi (2017), Adami et al. (2015a), Adami et al. (2015b), Egilmez e Tekalp (2014) e Silva et al. (2013). De modo similar, nenhum componente externo ao *gateway* multimídia é empregado para realizar a inspeção e classificação dos fluxos multimídia, para a realização de interações com o controlador OpenFlow, bem como para o fornecimento dos recursos de cache na rede local, tal como ocorre nos trabalhos de Adami et al. (2015b), Karaman et al. (2015), Chiang e Li (2016), Trajano e Fernandez (2016) e Georgopoulos et al. (2015). Por sua vez, a identificação e classificação de fluxos multimídia é realizada de modo distinto aos trabalhos de Seddiki et al. (2015), Seddiki et al. (2014) e Karaman et al. (2015), sendo realizada por meio do campo DSCP do protocolo IP, tal como nos trabalhos de Amelyanovich et al. (2017), López-Rodríguez e Campelo (2014) e Silva et al. (2013), porém com escopo e organização de *bits* enfatizando a identificação de múltiplos serviços multimídia, tais como serviços de áudio, de voz, de vídeo e de dados, bem como seus respectivos canais de comunicação. Além disso, a proposta não considera mapeamentos L2/L3 no *gateway* com suporte ao OpenFlow (nesse caso, *gateway* multimídia), tal como em Silva et al. (2013), bem como o emprego de outros protocolos no plano de dados, tal como em López-Rodríguez e Campelo (2014). Mecanismos de escalonamento, tal como em Ishimori et al. (2013), são empregados no âmbito da QoS, porém não como uma extensão ao protocolo OpenFlow, mas como parte dos recursos funcionais do *gateway* multimídia. Quanto ao emprego de recursos de predição, de modo similar aos trabalhos de Kryftis et al. (2016a), Kryftis et al. (2016b) e Costa Filho et al. (2016), este trabalho também discorre sobre o emprego de recursos de predição no âmbito de comunicações multimídia. Por outro lado, a abordagem proposta é distinta quanto ao método de predição empregado (nesse caso, baseado no filtro de Kalman e em seu subfiltro), bem como quanto aos indicadores de rede observados e utilizados durante o processo de predição (nesse caso, baseado em identificadores de serviços multimídia associados aos fluxos de tráfego multimídia da rede, tal como por meio de identificadores DSCP em redes IP, por exemplo).

## 2.5 – Síntese e Encerramento do Capítulo

Este Capítulo abordou os principais conceitos e fundamentos que embasam esta Tese, bem como discutiu, de forma não exaustiva, alguns dos trabalhos que estão relacionados com a mesma. Para tal, no âmbito dos conceitos e fundamentos que embasam esta Tese, questões envolvendo comunicações multimídia (com ênfase quanto às aplicações multimídia, aos protocolos multimídia e à distribuição de conteúdos multimídia na Internet), modelos de predição (com ênfase quanto ao filtro de Kalman e ao subfiltro de Kalman) e Redes Definidas por *Software* (com ênfase quanto à arquitetura, principais componentes e princípios básicos das SDNs e quanto ao protocolo OpenFlow) foram discutidos de modo a contextualizar o cenário ao qual este trabalho foi proposto e realizado. De modo complementar a tais conceitos e fundamentos, alguns dos trabalhos relacionados com esta Tese foram discutidos no âmbito da mesma, possibilitando ao leitor identificar e distinguir as abordagens empregadas em tais trabalhos das abordagens empregadas nesta Tese, nesse caso, no âmbito de um *gateway* multimídia em ambientes SDN OpenFlow e com ênfase no encaminhamento, na distribuição e na predição de conteúdos multimídia.

Nesse contexto, o próximo Capítulo discorre sobre o *gateway* multimídia em ambientes SDN OpenFlow, abordando, em linhas gerais, os conceitos e fundamentos quanto ao mesmo no âmbito desta Tese.

### 3 – Gateway Multimídia em Ambientes SDN OpenFlow

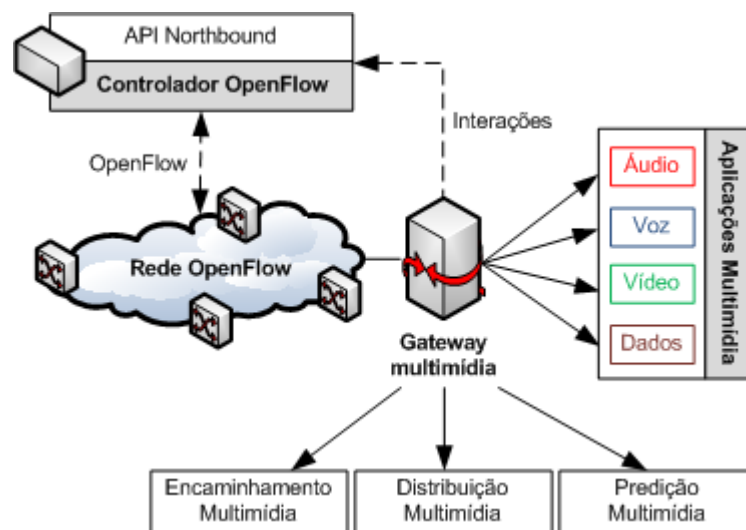
Este Capítulo tem o objetivo de apresentar ao leitor um *gateway* com recursos multimídia, tido como *gateway* multimídia, enquanto plataforma para aprimorar comunicações multimídia no âmbito de Redes Definidas por *Software* OpenFlow. Para tal, a primeira Seção, que trata de uma visão geral sobre o *gateway* multimídia em ambientes SDN OpenFlow, apresenta alguns conceitos e fundamentos quanto ao mesmo em tal cenário de rede; a segunda Seção discorre sobre a abordagem empregada junto ao *gateway* multimídia para a identificação e classificação de fluxos de tráfego multimídia na rede, com ênfase quanto ao emprego de identificadores de serviços multimídia, por meio de rótulos DSCP, para tal finalidade; a terceira Seção discorre sobre a arquitetura e os recursos funcionais propostos para o *gateway* multimídia, com ênfase quanto ao encaminhamento, à distribuição e à predição de conteúdos multimídia na rede; a quarta Seção discorre sobre um elemento intermediário, tido como adaptador de tráfego multimídia, enquanto solução complementar para a rotulação de fluxos de tráfego multimídia na rede; e, por sua vez, a quinta Seção discorre sobre algumas considerações quanto ao *gateway* multimídia no âmbito de comunicações multimídia.

#### 3.1 – Visão Geral

Nesta Tese, um *gateway* multimídia é tido como um elemento de rede capaz de fornecer recursos para aprimorar comunicações multimídia no ambiente ao qual está inserido. Em linhas gerais, seus recursos e funcionalidades são voltados e especializados quanto ao encaminhamento e à distribuição de conteúdos multimídia na rede, além de proporcionar um método de predição por meio de informações sobre tais conteúdos, de modo geral. Para tal, é capaz de distinguir e fornecer tratamento diferenciado para fluxos de tráfego associados a serviços multimídia, tais como para os serviços de áudio, de voz, de vídeo ou de dados, por exemplo. Nesse cenário, essa identificação/distinção de tráfego e seu encaminhamento à aplicação receptora devem ser realizados em tempo real, considerando o tipo de tráfego em questão (*stream* ou elástico, por exemplo), bem como o ambiente de rede ao qual esse tráfego está sendo gerado e consumido (ambiente com um ou com vários transmissores e receptores multimídia e empregando uma ou múltiplas aplicações multimídia, por exemplo).

Diante desse cenário, no âmbito de Redes Definidas por *Software*, o *gateway* multimídia atua como elemento complementar aos dispositivos do plano de dados e ao controlador OpenFlow da SDN, tido como *gateway* de acesso à rede para os sistemas finais do ambiente ao qual está inserido. Nesse cenário, objetiva estender as características de gerenciamento centralizado fornecidas pelo ambiente SDN para flexibilizar o encaminhamento e a distribuição de conteúdos multimídia na rede, bem como proporcionar um método de predição com base em informações sobre os fluxos de tráfego processados, nesse caso, com ênfase quanto à utilização de identificadores de serviços multimídia para essa finalidade. De modo complementar, também provê um arcabouço para a implementação de diferentes técnicas para encaminhamento seletivo de pacotes, de QoS e de cache multimídia, dentre outras. Nessa estrutura, conforme ilustrado na Figura 6, as interações entre o *gateway* multimídia e o controlador OpenFlow se dão por meio de uma extensão a API *Northbound* do controlador, tal como por meio uma API REST, por exemplo.

**Figura 6.** *Gateway* multimídia em ambientes SDN OpenFlow.

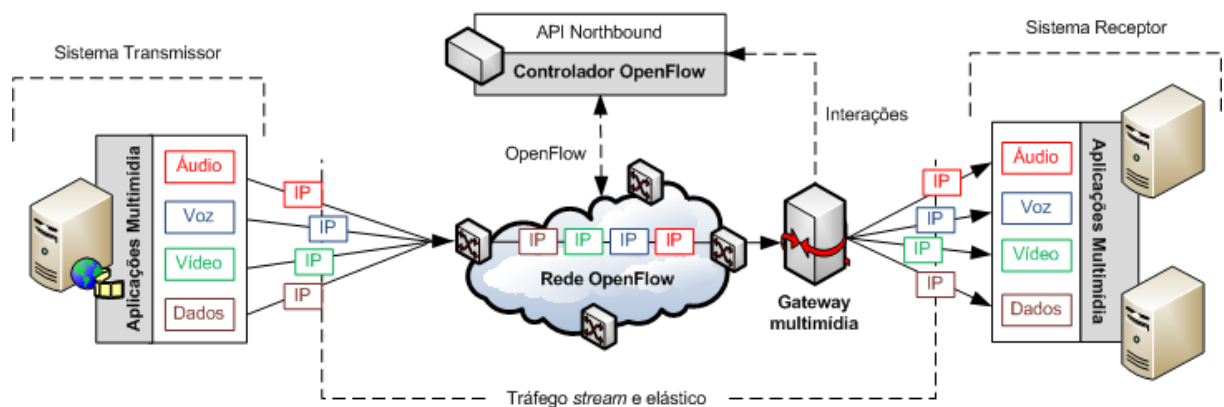


Em tal ambiente, múltiplas aplicações multimídia geram fluxos de tráfego tipicamente *stream* na rede. Esses fluxos de tráfego são intermediados pelo *gateway* multimídia e, após processados e tratados por ele em termos de seus recursos funcionais, encaminhados às suas respectivas aplicações receptoras (em execução em um ou mais clientes multimídia da rede). Nesse cenário, essa intermediação ocorre de modo transparente aos *hosts* e aplicações da rede, tal como por meio de um *switch* ou roteador convencional no plano de dados da SDN, por exemplo. Para tal, o *gateway* multimídia atua como “módulo

complementar” (externo e intermediário) dos sistemas transmissor e receptor do conteúdo multimídia, sendo posicionado (implementado) como *gateway* de acesso à rede para os clientes finais do ambiente SDN OpenFlow.

Nesse contexto, a Figura 7 ilustra, de modo geral, os sistemas transmissor e receptor multimídia, com o *gateway* multimídia, em um ambiente SDN OpenFlow.

**Figura 7.** Sistemas transmissor e receptor multimídia, com o *gateway* multimídia, em um ambiente SDN OpenFlow.



Em tal ambiente, pelo fato do *gateway* multimídia ser posicionado como *gateway* de acesso à rede dos sistemas finais, além de fluxos de tráfego associados a serviços multimídia, tais como fluxos de áudio, de voz ou de vídeo, por exemplo, ele pode receber outros fluxos de tráfego, tais como fluxos de transferência de arquivos ou de acesso a *e-mail*, dentre outros. Nesse caso, tais fluxos de tráfego podem ser tratados de modo distinto quanto aos fluxos multimídia ou, até mesmo, bloqueados/descartados pelo administrador da rede.

Diante desse cenário, por meio de interações com o controlador OpenFlow, o *gateway* multimídia possibilita que fluxos de tráfego multimídia, tais como fluxos de dados, de áudio, de voz ou de vídeo, por exemplo, recebam um tratamento diferenciado quanto a seu processamento e encaminhamento, tais como ser seletivamente encaminhado para um ou mais clientes multimídia, receber diferentes níveis de QoS por meio de classes de tráfego com configurações específicas de alocação de banda e de priorização, ou utilizar uma estrutura de cache para auxiliar na distribuição de seus conteúdos aos clientes finais. Em linhas gerais, essa abordagem é útil, por exemplo, para fornecer melhores níveis de QoS às aplicações multimídia com fluxo de tráfego do tipo *stream*, tais como pertinentes aos serviços multimídia



de voz ou de vídeo, dentre outros, bem como para aprimorar a percepção de QoE do usuário final no âmbito de tais aplicações.

### 3.2 – Identificação e Classificação de Fluxos de Tráfego Multimídia

Para distinguir e viabilizar o processamento e encaminhamento diferenciado de múltiplos fluxos de tráfego multimídia na rede, é necessária a utilização de mecanismos que, inicialmente, possibilitem a identificação e classificação de tais fluxos de tráfego de acordo com seu tipo de serviço multimídia (áudio, voz, vídeo ou dados, por exemplo).

Nesse contexto, no âmbito do protocolo OpenFlow, têm-se informações sobre fluxo desde a camada de enlace até a camada de transporte, com a possibilidade de combinação e utilização de múltiplos *match fields* para sua identificação e classificação (Figura 8). Em linhas gerais, tal como descrito anteriormente, essa abordagem possibilita múltiplas opções de gerência e de controle quanto aos fluxos de tráfego de rede, tais como no âmbito da engenharia de tráfego (MENDIOLA et al, 2017), da QoS (KARAKUS; DURRESI, 2017) e da segurança da rede (RAWAT; REDDY, 2017), dentre outras.

**Figura 8.** *Match fields* e estatísticas passíveis de utilização de acordo com as versões 1.0 até 1.4 do protocolo OpenFlow. Fonte: Kreutz et al. (2015).

OpenFlow Version	Match fields	Statistics	# Matches		# Instructions		# Actions		# Ports	
			Req	Opt	Req	Opt	Req	Opt	Req	Opt
v 1.0	Ingress Port	Per table statistics	18	2	1	0	2	11	6	2
	Ethernet: src, dst, type, VLAN	Per flow statistics								
	IPv4: src, dst, proto, ToS	Per port statistics								
	TCP/UDP: src port, dst port	Per queue statistics								
v 1.1	Metadata, SCTP, VLAN tagging	Group statistics	23	2	0	0	3	28	5	3
	MPLS: label, traffic class	Action bucket statistics								
v 1.2	OpenFlow Extensible Match (OXM)		14	18	2	3	2	49	5	3
	IPv6: src, dst, flow label, ICMPv6									
v 1.3	PBB, IPv6 Extension Headers	Per-flow meter	14	26	2	4	2	56	5	3
		Per-flow meter band								
v 1.4	—	—	14	27	2	4	2	57	5	3
		Optical port properties								

Nesse cenário, para efeitos de experimentação e avaliação dos recursos funcionais empregados junto ao *gateway* multimídia, optou-se pela utilização de rótulos (identificadores) DSCP (RFC 2474), fornecidos por meio dos campos ToS (*Type of Service*) no IPv4 e TC

(*Traffic Class*) no IPv6, para a identificação e classificação dos diferentes fluxos de tráfego multimídia da rede. Em linhas gerais, essa escolha foi embasada, em especial, pela possibilidade da utilização de tais rótulos enquanto possíveis identificadores de serviços multimídia na rede, além de já ter sido explorada em outros trabalhos já desenvolvidos pelo grupo de trabalho ao qual esta Tese está vinculada. Além disso, tal como descrito no Capítulo anterior, alguns trabalhos recentes (AMELYANOVICH et al., 2017; LÓPEZ-RODRÍGUEZ; CAMPELO, 2014; SILVA et al., 2013) já exploraram abordagens similares para a identificação de fluxos de tráfego de rede por meio de identificadores DSCP, mostrando a viabilidade de tal abordagem para essa finalidade. Ainda assim, é importante destacar que, mesmo considerando o uso de identificadores DSCP para a identificação e classificação de fluxos de tráfego multimídia na rede, a abordagem proposta não se restringe somente ao mesmo para tal, sendo passível de eventuais modificações sem comprometer a arquitetura e os recursos funcionais propostos para o *gateway* multimídia.

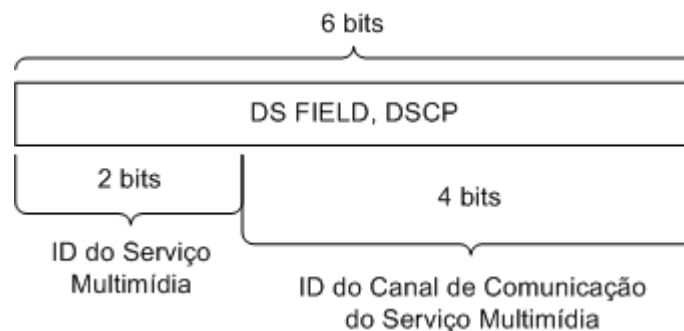
De modo complementar, a abordagem proposta também considera que os *bits* disponíveis junto ao campo DSCP (nesse caso, 6 *bits*) sejam organizados de modo a possibilitar, por exemplo, a identificação de possíveis canais de comunicação associados aos serviços multimídia da rede. Dessa forma, mesmo para serviços multimídia idênticos, tais como serviços multimídia de vídeo utilizando dois canais de comunicação distintos, por exemplo, diferentes níveis de QoS poderiam ser fornecidos para cada canal em particular, nesse caso, por meio do *gateway* multimídia no ambiente SDN OpenFlow.

Como exemplo, considerando apenas os serviços multimídia de dados, de áudio, de voz e de vídeo na rede, apenas 2 dos 6 *bits* fornecidos junto ao campo DSCP poderiam ser empregados para viabilizar a identificação de tais serviços multimídia, de modo que os 4 *bits* restantes poderiam ser organizados para identificar até 16 possíveis canais de comunicação para cada um desses 4 serviços multimídia em questão. Essa abordagem, além de explorar de uma melhor maneira o campo DSCP (evitando, por exemplo, o “desperdício” de utilização de seus *bits*), possibilita, no protocolo IPv4, a pseudo implementação do campo “*Flow Label*” presente no cabeçalho do protocolo IPv6, viabilizando a identificação de canais distintos para, também, distintos serviços multimídia por meio de tal campo (sem a necessidade de utilização de outros campos do cabeçalho do protocolo IP, inexistentes para esse propósito no IPv4).

Nesse cenário, a Figura 9 ilustra essa possível organização de *bits* para a identificação e classificação dos serviços multimídia de dados, de áudio, de voz e de vídeo na rede, bem como dos canais de comunicação associados aos mesmos, ambos por meio do

campo DSCP do protocolo IP. De modo complementar, a Tabela 1 apresenta os possíveis identificadores DSCP pertinentes a cada serviço multimídia em questão, bem como de seus respectivos canais de comunicação.

**Figura 9.** Possível organização de *bits* para a identificação e classificação dos serviços multimídia de dados, de áudio, de voz e de vídeo na rede, bem como dos canais de comunicação associados aos mesmos, ambos por meio do campo DSCP do protocolo IP.



**Tabela 1.** Possíveis identificadores DSCP para os serviços multimídia de dados, de áudio, de voz e de vídeo na rede, bem como de seus respectivos canais de comunicação.

Tipo de Serviço Multimídia	Identificador do Serviço Multimídia	Identificador do Canal de Comunicação do Serviço Multimídia
Dados	00	de "0000" até "1111"
Áudio	01	
Voz	10	
Vídeo	11	

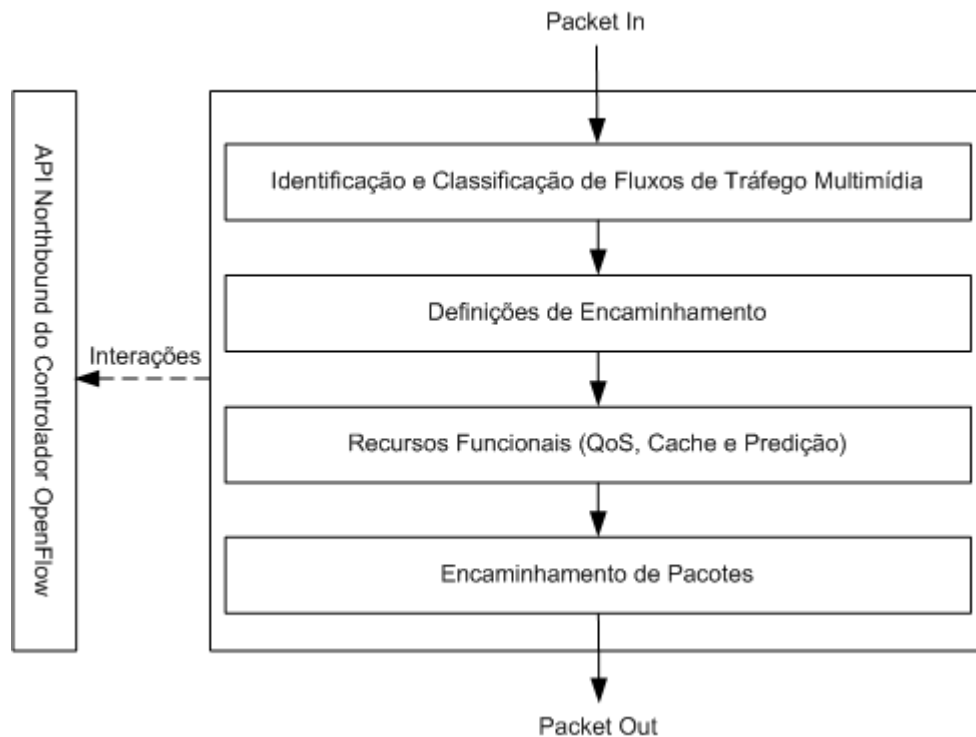
Nesse cenário, é importante destacar que essa organização de *bits* pode ser facilmente adequada para suportar mais ou menos serviços multimídia, aumentando-se ou diminuindo-se os *bits* utilizados para a identificação do serviço multimídia, bem como os *bits* utilizados para identificação dos canais de comunicação associados ao serviço multimídia em questão (respeitando-se o limite total de *bits* do campo DSCP).

### 3.3 – Arquitetura e Recursos Funcionais

Conforme descrito anteriormente, no ambiente SDN, o *gateway* multimídia atua como elemento complementar aos dispositivos do plano de dados e ao controlador OpenFlow, tido como *gateway* de acesso à rede para os sistemas finais do ambiente ao qual está inserido. Nesse cenário, múltiplas aplicações multimídia geram fluxos de tráfego tipicamente *stream* na rede. Esses fluxos de tráfego são intermediados pelo *gateway* multimídia e, após processados e tratados por ele em termos de seus recursos funcionais, encaminhados às suas respectivas aplicações receptoras (em execução em um ou mais clientes multimídia da rede).

Nesse contexto, a Figura 10 ilustra, de modo geral, a arquitetura de alto nível do *gateway* multimídia no ambiente SDN OpenFlow.

**Figura 10.** Visão geral da arquitetura de alto nível do *gateway* multimídia no ambiente SDN OpenFlow.



Em tal arquitetura, todo fluxo de tráfego intermediado pelo *gateway* multimídia é identificado e classificado de acordo com seu tipo de serviço multimídia, em especial, por meio de identificadores de serviços multimídia via rótulos DSCP, tal como descrito na Seção anterior. Na eventual impossibilidade de identificação de tais rótulos, outros critérios e

estratégias podem ser empregados, em ambos os casos, por meio dos diferentes *match fields* fornecidos pelo protocolo OpenFlow. Nesse cenário, fluxos tidos como “não identificados”, tais como pela ausência de critérios para sua identificação e classificação, por exemplo, serão processados de acordo com a definição padrão empregada junto ao *gateway* multimídia, tais como por meio de uma regra de fluxo padrão (desconsiderando a oferta de diferentes níveis de QoS, por exemplo), do descarte de seus pacotes ou do envio dos mesmos ao controlador para que uma ação específica seja executada.

Após esse processo, os diferentes fluxos de tráfego multimídia podem ser encaminhados aos clientes da rede de acordo com regras de fluxos específicas para tal. Em linhas gerais, essas regras compreendem, em especial, encaminhamentos seletivos de pacotes, redirecionamento de pacotes ou descarte/bloqueio de pacotes, ambas no âmbito das possibilidades proporcionadas pelo protocolo OpenFlow. Nesse contexto, o encaminhamento seletivo de pacotes possibilita a escolha de qual (ou quais) fluxo(s) serão encaminhados pelo *gateway* multimídia. Dessa forma, as definições de encaminhamento ficam especializadas em termos de um ou mais fluxos multimídia, havendo a possibilidade de realização de descartes ou processamento padrão de pacotes para os demais fluxos de tráfego. Por sua vez, no redirecionamento de pacotes, os pacotes IP de um ou mais fluxos multimídia podem ter seu destinatário modificado, sendo passíveis de redirecionamentos para outros *hosts* e aplicações de rede. Nesse caso, há a possibilidade de se modificar tanto o IP do destinatário (e, se aplicável, do remetente), quanto o número da porta de comunicação da aplicação multimídia, possibilitando redirecionamento nos níveis de rede e de transporte. Adicionalmente, tais recursos também podem ser empregados aos demais fluxos de tráfego da rede, nesse caso, utilizando os diferentes *match fields* proporcionados pelo protocolo OpenFlow (considerando, por exemplo, que tais fluxos não possuem identificadores de serviços multimídia via DSCP).

Nesse cenário, as interações entre o *gateway* multimídia e o controlador OpenFlow se dão por meio de uma extensão a API *Northbound* do controlador, tal como por meio uma API REST, por exemplo. Por meio dessas interações, as definições de encaminhamento escritas junto ao *gateway* multimídia, tal como por meio de *shell scripts*, por exemplo, podem ser especificadas junto ao controlador OpenFlow. Por sua vez, via mensagens OpenFlow, tais como para a inserção ou remoção de regras de fluxos, por exemplo, essas definições são efetivamente implementadas no *gateway* multimídia, viabilizando o processamento e encaminhamento diferenciado dos fluxos de tráfego associados a tais regras de fluxos, de modo geral.

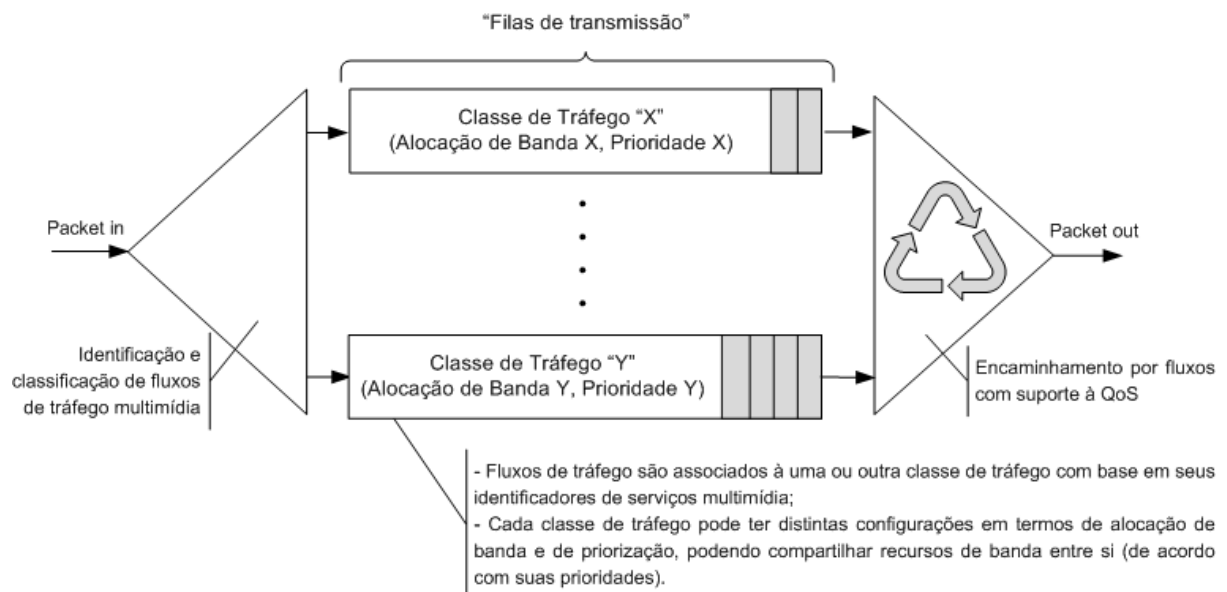
Quanto ao fornecimento de diferentes níveis de QoS durante o encaminhamento dos fluxos de tráfego pelo *gateway* multimídia, múltiplas classes de tráfego, com distintas configurações em termos de alocação de banda e de priorização podem ser empregadas junto ao mesmo. Em linhas gerais, essas classes de tráfego são especialmente fornecidas por meio de diferentes escalonadores de pacotes junto ao *gateway* multimídia, tais como HTB, SFQ e CBQ (*Class Based Queueing*), dentre outros. Em um contexto mais amplo, essa abordagem possibilita a implementação de múltiplas filas de transmissão, com definições de alocação de banda e de priorização distintas entre si. Nesse cenário, de modo a maximizar a utilização dos recursos de rede, uma ou mais classes de tráfego podem compartilhar recursos de banda entre si, de modo que seja possível, por exemplo, que o escopo inicialmente definido para os fluxos de uma determinada classe de tráfego possa ser ultrapassado considerando as disponibilidades de banda de outra classe de tráfego. Nesse caso, esse “empréstimo de recursos” considera a prioridade de cada classe de tráfego em questão, de modo que classes de maior prioridade sejam beneficiadas antecipadamente quando comparadas às classes de menor prioridade.

Dessa forma, distintas configurações no âmbito da QoS podem ser fornecidas para cada fluxo de tráfego em questão, em ambos os casos, pela classificação dos diferentes fluxos da rede, por meio de seus identificadores de serviços multimídia, com sua respectiva classe de tráfego junto ao *gateway* multimídia. Nesse cenário, na ausência de eventuais identificadores de serviços multimídia, bem como de eventuais definições quanto à QoS, uma classe de tráfego padrão pode ser utilizada para acomodar um ou mais fluxos de tráfego, porém, sem a oferta diferenciada de QoS para ambos. Em linhas gerais, essa abordagem possibilita que fluxos de tráfego específicos, tais como fluxos *stream* pertinentes aos serviços multimídia de voz ou vídeo, por exemplo, possam receber níveis de QoS distintos dos demais fluxos de tráfego da rede. Além disso, considerando a identificação de diferentes canais de comunicação para cada fluxo de tráfego em questão, ambos por meio dos identificadores de serviços multimídia pertinentes aos mesmos, fluxos associados ao mesmo serviço multimídia, porém empregando distintos canais de comunicação, também poderiam receber distintas configurações no âmbito da QoS. Essa abordagem é útil, por exemplo, para fornecer uma melhor percepção de QoE ao usuário final quanto aos serviços multimídia (isto é, fluxos de tráfego pertinentes aos serviços multimídia) que terão tratamento diferenciado no âmbito da QoS junto ao *gateway* multimídia.

Nesse contexto, a Figura 11 ilustra, de modo geral, essa abordagem para a oferta de diferentes níveis de QoS junto ao *gateway* multimídia, nesse caso, baseada no emprego de

múltiplas classes de tráfego, com definições de alocação de banda e de priorização distintas entre si, e classificações de tráfego por meio de identificadores de serviços multimídia.

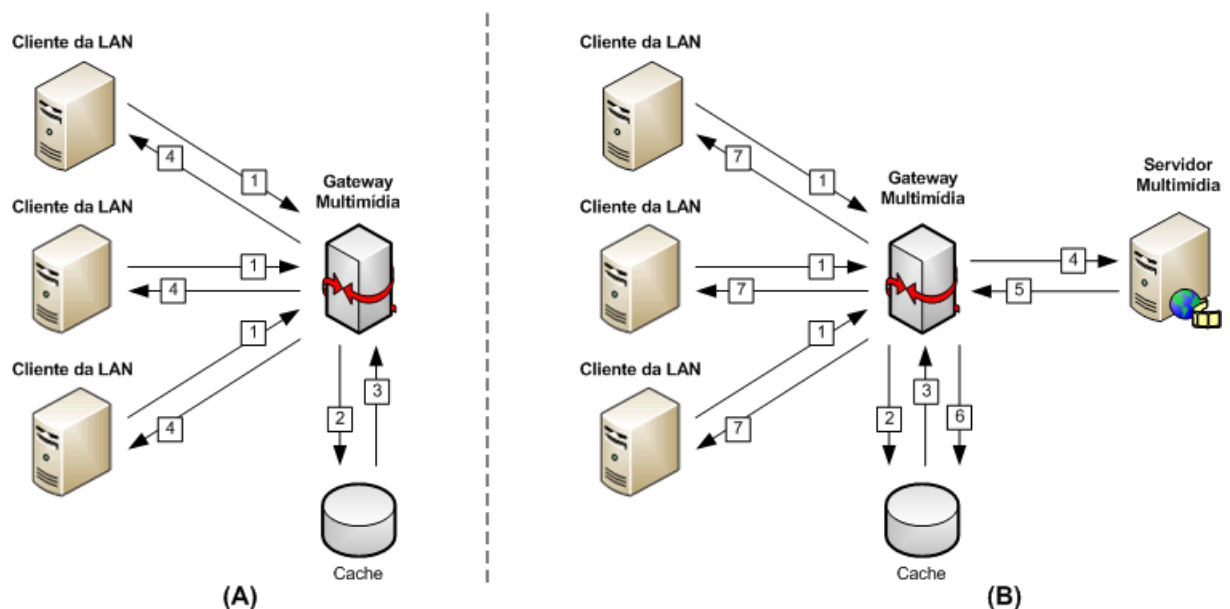
**Figura 11.** Visão geral quanto à oferta de diferentes níveis de QoS junto ao *gateway* multimídia: abordagem baseada no emprego de múltiplas classes de tráfego, com definições de alocação de banda e de priorização distintas entre si, e classificações de tráfego por meio de identificadores de serviços multimídia.



De modo complementar a essa abordagem no âmbito do encaminhamento por fluxos com suporte à QoS, a proposta também considera que recursos de cache auxiliem na distribuição dos conteúdos multimídia aos clientes da rede. Para tal, o *gateway* multimídia pode atuar como Proxy para os clientes da rede local, centralizando as requisições realizadas por tais clientes e fornecendo uma estrutura de cache para aproximar o conteúdo multimídia previamente requisitado do ambiente de rede ao qual ele (*gateway* multimídia) está inserido. Nesse caso, durante o processo de entrega do conteúdo multimídia para os clientes da rede, uma cópia do conteúdo requisitado pode ser armazenada na estrutura de cache do *gateway* multimídia. Em futuras requisições, o *gateway* multimídia verifica se o conteúdo requisitado possui uma cópia em cache e, se existir, recupera e distribui tal conteúdo diretamente do mesmo, sem utilizar, por exemplo, recursos de Internet para tal. Caso contrário, o conteúdo requisitado é recuperado do servidor multimídia na Internet, armazenado em cache (para futuras requisições) e, em seguida, entregue ao cliente final.

Nesse contexto, como exemplo, a Figura 12 ilustra dois cenários contemplando a utilização de recursos de cache no âmbito da distribuição de conteúdos multimídia aos clientes da rede: um cenário na qual o conteúdo multimídia é recuperado e distribuído diretamente do cache aos clientes da rede (cenário A), nesse caso, sem utilizar recursos de Internet, e um cenário na qual o conteúdo multimídia é recuperado do servidor multimídia na Internet, armazenado em cache e, em seguida, distribuído aos clientes da rede (cenário B).

**Figura 12.** Visão geral quanto à oferta de recursos de cache junto ao *gateway* multimídia: conteúdo multimídia recuperado e distribuído diretamente do cache aos clientes da rede (cenário A) e conteúdo multimídia recuperado do servidor multimídia na Internet, armazenado em cache e distribuído aos clientes da rede (cenário B).



Em tais cenários ilustrados por meio da Figura 12, múltiplos clientes finais podem requisitar conteúdos multimídia, tais como conteúdos de vídeo, por exemplo, em seu ambiente de rede local (etapa 1, cenários A e B). Ao intermediar tais requisições, atuando como Proxy no ambiente SDN, o *gateway* multimídia verifica se o conteúdo requisitado encontra-se em sua estrutura de cache (etapas 2 e 3, cenários A e B) e, se existir, entrega o conteúdo em questão para seus respectivos clientes finais (etapa 4, cenário A). Caso contrário, o *gateway* multimídia obtém o conteúdo requisitado junto ao servidor multimídia na Internet (etapas 4 e 5, cenário B), armazena-o em sua estrutura de cache (etapa 6, cenário B) para, em seguida, entregá-lo aos seus respectivos clientes finais (etapa 7, cenário B). De modo

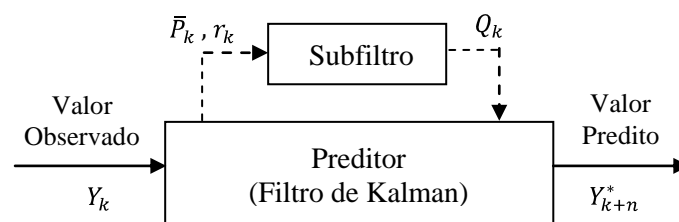


complementar, observe, no cenário B da Figura 12, que as múltiplas requisições dos clientes da rede local são agregadas em uma única requisição do *gateway* multimídia na recuperação do conteúdo solicitado na Internet (isto é, no servidor multimídia na Internet). Dessa forma, além de aproximar o conteúdo previamente requisitado dos clientes da rede local, com contribuições no âmbito da QoS e da QoE, possibilitando, por exemplo, melhores taxas e tempos de transmissão na segunda e demais requisições pertinentes ao conteúdo multimídia em questão, essa abordagem também é útil para amenizar a utilização dos enlaces de rede entre os clientes multimídia e o ISP na qual o servidor multimídia na Internet está conectado, minimizando, por exemplo, eventuais congestionamentos, atrasos e perdas de pacotes durante esse processo de recuperação e distribuição de conteúdos multimídia, de modo geral.

Por sua vez, a arquitetura proposta também considera que essa abordagem baseada em identificadores de serviços multimídia possa ser complementada por um método de predição no âmbito da distribuição multimídia. Para tal, a partir de observações sobre os fluxos de tráfego já processados pelo *gateway* multimídia (isto é, observações sobre os identificadores de serviços multimídia pertinentes aos fluxos de tráfego já processados pelo *gateway* multimídia), a proposta considera o emprego de um módulo preditor, baseado no filtro de Kalman e no subfiltro de Kalman, para que futuros conteúdos multimídia (isto é, futuros fluxos com conteúdos multimídia, tais como pertinentes aos serviços de áudio, de voz, de vídeo ou de dados, por exemplo) possam ser preditos na rede.

Nesse contexto, como exemplo, a Figura 13 ilustra, de modo geral, a estrutura funcional quanto a tal módulo preditor de conteúdos multimídia, nesse caso, baseado em identificadores de serviços multimídia e fornecido pelo filtro de Kalman e pelo subfiltro de Kalman.

**Figura 13.** Visão geral da estrutura funcional do módulo preditor de conteúdos multimídia, nesse caso, baseado em identificadores de serviços multimídia e fornecido por meio do filtro de Kalman e do subfiltro de Kalman.



Em tal estrutura,  $Y_k$  corresponde aos identificadores de serviços multimídia observados nos fluxos de tráfego já processados pelo *gateway* multimídia;  $\bar{P}_k$  refere-se à covariância do erro do estado estimado;  $r_k$  refere-se ao erro de predição;  $Q_k$  refere-se à covariância de ruído; e, por sua vez,  $Y_{k+n}^*$  corresponde ao valor filtrado/predito (isto é, fluxos de tráfego com conteúdos multimídia filtrados/preditos, ambos por meio de seus respectivos identificadores de serviços multimídia) pelo filtro de Kalman e pelo subfiltro de Kalman, nesse caso, por meio de suas respectivas equações descritas nas Seções 2.2.1 e 2.2.2.

### 3.4 – Adaptador de Tráfego Multimídia

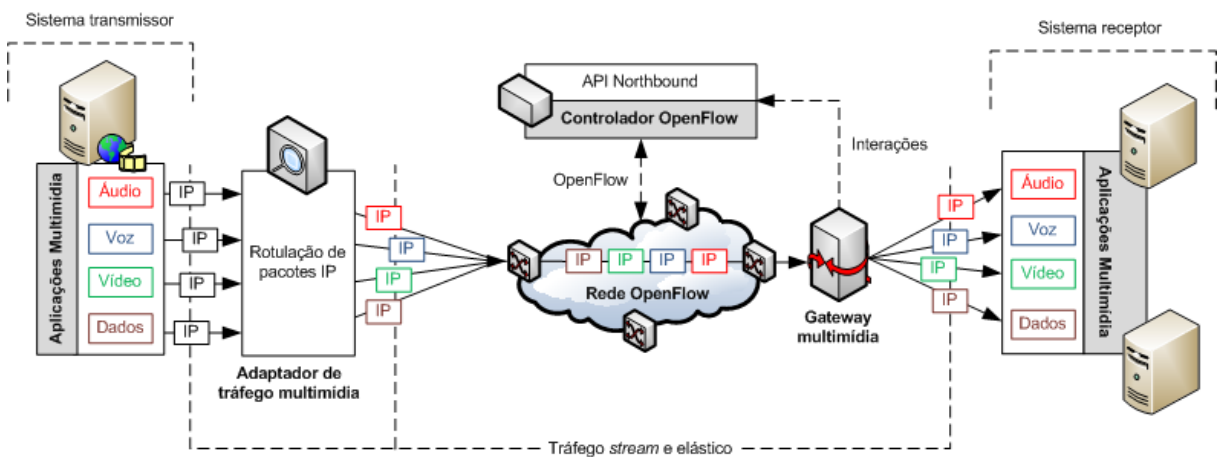
Conforme descrito anteriormente, propõe-se que o *gateway* multimídia utilize, em especial, identificadores de serviços multimídia para viabilizar seus recursos funcionais. Em linhas gerais, esses identificadores são obtidos a partir dos fluxos de tráfego multimídia intermediados e processados pelo *gateway* multimídia no ambiente de rede ao qual ele está inserido, nesse caso, por meio dos diferentes *match fields* proporcionados pelo protocolo OpenFlow, em especial, via rótulos DSCP.

Nesse cenário, no que se refere ao campo DSCP do protocolo IP, especialmente explorado e empregado na experimentação e avaliação do *gateway* multimídia, é necessário que os fluxos de tráfego multimídia da rede estejam previamente rotulados com tais identificadores antes de serem efetivamente processados pelo *gateway* multimídia. Esse processo de rotulação pode ocorrer, preferencialmente, pela própria aplicação transmissora do conteúdo multimídia ou por meio de um elemento intermediário aos sistemas transmissor e receptor do conteúdo em questão, tal como por meio de um *switch* ou roteador de ingresso à rede, por exemplo. Independentemente do método utilizado, deve-se ter uma coerência entre os identificadores presentes nos fluxos de tráfego multimídia e na configuração (identificação e classificação) desses junto ao *gateway* multimídia, de modo que uma “interpretação incorreta”, pertinente a uma definição distinta de identificadores nas aplicações e no *gateway* multimídia, por exemplo, pode inviabilizar as transmissões intermediadas pelo mesmo.

Diante desse cenário, esta Tese considera o emprego de um elemento intermediário aos sistemas transmissor e receptor do conteúdo multimídia, tido como “adaptador de tráfego multimídia” (Figura 14), para viabilizar esse processo de rotulação de pacotes de acordo com seus respectivos identificadores de serviços multimídia, nesse caso,

por meio do campo DSCP do protocolo IP. Em linhas gerais, essa abordagem baseia-se, em especial, na impossibilidade de se definir tais identificadores diretamente nos serviços e aplicações utilizados na experimentação e avaliação do *gateway* multimídia (seja por não se ter acesso aos códigos-fonte de tais serviços e aplicações ou, nesse momento, não serem objetivos desta Tese, mas sim um meio para experimentar e avaliar a abordagem proposta na mesma).

**Figura 14.** Sistemas transmissor e receptor do conteúdo multimídia, com o adaptador de tráfego multimídia e o *gateway* multimídia, em um ambiente SDN OpenFlow.



Nesse cenário, cabe ao adaptador de tráfego multimídia realizar o processo de rotulação de pacotes de acordo com seus respectivos identificadores de serviços multimídia, nesse caso, por meio do campo DSCP do protocolo IP. Dessa forma, após devidamente rotulados, esses pacotes e seus fluxos de tráfego podem receber um tratamento diferenciado ao longo da rede, seja pelo *gateway* multimídia ou pelos demais elementos do plano de dados da SDN. Ainda assim, é importante destacar que esses identificadores de serviços multimídia, assim como outros possíveis *match fields* associados aos seus respectivos fluxos de tráfego, não significam, necessariamente, que tais pacotes receberão tratamento diferenciado ao longo da rede, mas sim que poderão ser identificados e classificados por meio dessa abordagem e, a partir dela, receberem esse eventual tratamento diferenciado, de modo geral.

### 3.5 – Algumas Considerações

Conforme brevemente descrito na Seção 2.1, uma série de questões e desafios estão relacionadas aos diversos fatores e particularidades que envolvem comunicações multimídia, com inúmeras discussões, abordagens e contribuições no âmbito de tal tema.

Diante desse cenário, no que se refere à abordagem empregada nesta Tese quanto ao fornecimento de recursos e funcionalidades para aprimorar comunicações multimídia no âmbito de um *gateway* multimídia em SDN, é importante ressaltar que tais recursos e funcionalidades estão restritos ao ambiente de rede local na qual o *gateway* multimídia está inserido. Dessa forma, em um processo de comunicação multimídia interdomínios ou no âmbito fim-a-fim na Internet, por exemplo, esses recursos e funcionalidades serão especialmente percebidos em seu ambiente de rede local e não, necessariamente, em todos os ambientes de rede entre os sistemas transmissor e receptor do conteúdo multimídia. Como exemplo, no âmbito da QoS, o *gateway* multimídia pode ser utilizado para priorizar, por meio da oferta de diferentes níveis de QoS na rede, os fluxos de tráfego do tipo *stream* dos demais fluxos de tráfego da rede. Nesse caso, essa priorização será especialmente perceptível no ambiente de rede na qual o *gateway* multimídia está inserido, uma vez que ambientes de rede distintos podem empregar abordagens também distintas quanto à priorização de seus fluxos de tráfego (de acordo com critérios técnicos, organizacionais e até comerciais desse outro ambiente, por exemplo).

Além disso, é importante destacar que apenas no âmbito da QoS há uma grande quantidade de fatores e características relacionadas à mesma, tais como eventuais exigências quanto à recursos de banda e de temporização que, dada à demanda pela aplicação ou serviço multimídia, podem, por si só, representar grande parte dos desafios voltados à distribuição de seus conteúdos. Um exemplo pertinente a tal contexto é descrito por Kurose e Ross (2010), na qual os autores discorrem que se um determinado provedor de IPTV desejasse transmitir um importante evento esportivo, como uma partida de Copa do Mundo, de um único servidor da Internet para cerca de 100 milhões de clientes com uma velocidade de 1 Mbps, a banda passante exigida pelo servidor seria de 100 Tbps, requisito que inviabilizaria tal transmissão num contexto servidor-cliente, por exemplo. Perceba que, nesse caso, antes mesmo de questões voltadas ao fornecimento de recursos de QoS ao conteúdo multimídia em si, uma série de outros fatores, tais como o dimensionamento e configuração dos enlaces do ISP, a

utilização ou não de CDNs, o emprego ou não de protocolos e técnicas específicas para a transmissão do conteúdo multimídia, a plataforma com a qual o conteúdo multimídia será acessado e disponibilizado aos clientes do ISP, dentre outros, são fatores cruciais relacionados ao processo de comunicação multimídia.

Nesse contexto, é importante destacar que a proposta quanto ao *gateway* multimídia objetiva contribuir com a melhoria desse cenário, agregando recursos que, combinados com outras técnicas e abordagens, são importantes para o aprimoramento desse processo de comunicação multimídia. Dessa forma, dada a amplitude do tema, o *gateway* multimídia representa uma pequena e importante parte nesse processo de comunicação, mas, por si só, não é a solução inovadora que resolverá os vários desafios relacionados ao tema, ainda sendo necessária a utilização de outras técnicas e abordagens que, em conjunto, objetivem e forneçam recursos de modo a aprimorar, em linhas gerais, esse processo de comunicação multimídia.

### **3.6 – Síntese e Encerramento do Capítulo**

Este Capítulo abordou um *gateway* com recursos multimídia, tido como *gateway* multimídia, enquanto plataforma para aprimorar comunicações multimídia no âmbito de Redes Definidas por *Software* OpenFlow. Para tal, apresentou uma visão geral quanto ao *gateway* multimídia em tal cenário de rede, apresentando alguns conceitos e fundamentos quanto ao mesmo, bem como tratando de questões envolvendo a identificação e classificação de fluxos de tráfego multimídia (com ênfase quanto ao emprego de identificadores de serviços multimídia, por meio de rótulos DSCP, para tal finalidade), de sua arquitetura e recursos funcionais (com ênfase quanto ao encaminhamento, à distribuição e à predição de conteúdos multimídia) e de um elemento intermediário, tido como adaptador de tráfego multimídia, enquanto solução complementar para a rotulação de fluxos de tráfego multimídia na rede. De modo complementar, este Capítulo também apresentou algumas considerações quanto ao *gateway* multimídia no âmbito de comunicações multimídia, de modo geral.

Nesse contexto, o próximo Capítulo discorre sobre os experimentos e resultados de avaliação pertinentes ao *gateway* multimídia no âmbito SDN OpenFlow, com ênfase no encaminhamento, na distribuição e na predição de conteúdos multimídia.

## 4 – Experimentos e Resultados

Este Capítulo discorre sobre os experimentos e resultados de avaliação pertinentes ao *gateway* multimídia no âmbito SDN OpenFlow. Para tal, a primeira Seção apresenta uma visão geral sobre os ambientes e ferramentas de experimentação e avaliação empregadas nesta Tese, nesse caso, no âmbito da emulação e da simulação; e, por sua vez, a segunda Seção discorre sobre os resultados obtidos por meio de tais ambientes, com ênfase quanto à rotulação de fluxos de tráfego multimídia, à identificação e encaminhamento diferenciado de fluxos de tráfego multimídia, às características de tráfego, tempos de transmissão, variações do atraso e perda de pacotes, ao encaminhamento por fluxos com suporte à QoS, ao cache de conteúdos multimídia e à predição de conteúdos multimídia.

### 4.1 – Ambientes e Ferramentas de Experimentação e Avaliação

Para a obtenção de resultados experimentais, dois ambientes de avaliação foram empregados: um ambiente de emulação e um ambiente de simulação. No âmbito do *hardware*, tais ambientes foram configurados em um computador com processador Intel Core i5-7200U 2.50 GHz, memória RAM de 8 GB (*single channel* DDR4 2133 MHz), placa de vídeo NVIDIA GeForce 940MX de 4 GB e interface de rede cabeada Gigabit Ethernet (10/100/1000).

Diante desse cenário, o ambiente de emulação foi utilizado para a experimentação e avaliação do *gateway* multimídia em um cenário adequado ao qual ele é proposto, ou seja, um cenário SDN OpenFlow. Por sua vez, o ambiente de simulação foi utilizado, em especial, para avaliar o método de predição de conteúdos multimídia em um cenário genérico, porém considerando o emprego de identificadores de serviços multimídia nesse processo, tal como no ambiente de emulação, com ênfase quanto à implementação matemática do módulo preditor por meio do filtro de Kalman e do subfiltro de Kalman em tal ambiente.

Nesse contexto, quanto às ferramentas utilizadas para viabilizar a implementação de tais ambientes de experimentação e avaliação, o ambiente de emulação foi configurado a

partir de um ambiente virtual, fornecido pela solução Oracle VM VirtualBox<sup>24</sup> em execução sobre o sistema operacional Linux Ubuntu<sup>25</sup>, e embasada nas soluções de *software* Open vSwitch<sup>26</sup> como *software switch*, Floodlight<sup>27</sup> como controlador SDN e por meio do protocolo OpenFlow. Por sua vez, o ambiente de simulação foi configurado por meio da solução de *software* Wolfram Mathematica<sup>28</sup>.

Em tais ambientes, diferentes versões de *softwares* foram utilizadas no decorrer do desenvolvimento desta Tese, de modo que as últimas experimentações pertinentes ao ambiente de emulação foram embasadas na versão 16.04 LTS do Linux Ubuntu, versão 5.1.18 do Oracle VM VirtualBox, versão 2.5.2 do Open vSwitch, versão 1.2 do Floodlight e versão 1.3 do protocolo OpenFlow. Por sua vez, as últimas experimentações pertinentes ao ambiente de simulação foram embasadas na versão 11 do *software* Wolfram Mathematica.

Nesse contexto, no que se refere ao ambiente de emulação, um protótipo pertinente ao *gateway* multimídia foi implementado como um *host* virtual por meio do Oracle VM VirtualBox. Em tal *host*, configurado sobre o sistema operacional Linux Ubuntu 14.04 LTS, sua integração e interoperação com a infraestrutura SDN foi realizada por meio do Open vSwitch versão 2.0.2 e do protocolo OpenFlow versão 1.3.

Em tal protótipo, quanto à implementação dos recursos experimentados e avaliados no ambiente de emulação, algumas ferramentas e soluções especialmente fornecidas pelo ambiente de *software* disponibilizado pelo Linux Ubuntu foram empregadas para tal. Dentre tais ferramentas, cabe destacar as soluções netfilter/iptables<sup>29</sup>, *tc* (por meio do pacote iproute2<sup>30</sup>) e Squid<sup>31</sup>, bem como de uma série de *shell scripts* construídos sobre o *shell* Bash do Linux. Nesse cenário, as interações entre o *gateway* multimídia e o controlador SDN foram realizadas por meio das *northbound* REST APIs fornecidas pelo Floodlight<sup>32</sup>, em especial, por meio de suas *Static Entry Pusher APIs*<sup>33</sup> no âmbito da inserção, remoção e listagem de regras de fluxos, dentre outros.

---

<sup>24</sup> Oracle VM VirtualBox em <https://www.virtualbox.org>

<sup>25</sup> Linux Ubuntu em <https://www.ubuntu.com>

<sup>26</sup> Open vSwitch em <http://openvswitch.org>

<sup>27</sup> Floodlight em <http://www.projectfloodlight.org/floodlight>

<sup>28</sup> Wolfram Mathematica em <https://www.wolfram.com/mathematica>

<sup>29</sup> netfilter/iptables em <http://www.netfilter.org>

<sup>30</sup> iproute2 em <http://www.linuxfoundation.org/collaborate/workgroups/networking/iproute2>

<sup>31</sup> Squid em <http://www.squid-cache.org>

<sup>32</sup> Floodlight REST API em

<https://floodlight.atlassian.net/wiki/display/floodlightcontroller/Floodlight+REST+API>

<sup>33</sup> Floodlight Static Entry Pusher API em

<https://floodlight.atlassian.net/wiki/display/floodlightcontroller/Static+Entry+Pusher+API>

Diante desse cenário, quanto aos conteúdos e fluxos de tráfego multimídia utilizados para a obtenção dos resultados experimentais, tanto no ambiente de emulação, quanto no ambiente de simulação, quatro serviços multimídia foram considerados para tal: serviços multimídia de áudio, de voz, de vídeo e de dados.

Nesse contexto, no que se refere ao ambiente de emulação, a geração dos fluxos de tráfego pertinentes a tais serviços foi realizada por meio de um gerador de tráfego multimídia com recursos de modelagem de tráfego (PINOTTI et al., 2011), da solução iPerf<sup>34</sup> e de três serviços de Internet: um serviço de *streaming* de áudio e vídeo (embasado nos protocolos RTP e RTSP), um serviço de transferência de arquivos (embasado no protocolo FTP) e um serviço de hospedagem *web* (embasado no protocolo HTTP).

Em tal ambiente, o gerador de tráfego multimídia possibilitou a geração de fluxos de tráfego de áudio, de voz, de vídeo e de dados na rede, ambos com modelagem de tráfego pré-definida e, nas experimentações, embasadas na função de distribuição exponencial. Por meio de tal recurso, também possibilitou a identificação de eventuais alterações quanto às características de seus fluxos de tráfego em termos de duração e de frequência de transmissão/recebimento de pacotes. Por outro lado, por não fornecer informações sobre tempos de transmissão, variações do atraso (*jitter*) e pacotes perdidos, a ferramenta iPerf foi empregada para tal. Por sua vez, os serviços de *streaming* de áudio e de vídeo, de transferência de arquivos e de hospedagem *web* foram utilizados, em especial, para avaliar os recursos de QoS e de cache em um contexto mais próximo ao âmbito do usuário final. De modo complementar, as ferramentas Wireshark<sup>35</sup>, ntopng<sup>36</sup> e MSU Quality Measurement Tool<sup>37</sup> foram empregadas, em especial, para questões voltadas à análise de fluxos de tráfego de rede e no âmbito da QoE percebida pelo usuário final, de modo geral.

Diante desse cenário, os serviços multimídia de áudio, de voz, de vídeo e de dados foram configurados a partir de dois servidores multimídia (*hosts* H1 e H2), tendo como clientes multimídia outros cinco *hosts* na rede (*hosts* H3, H4, H5, H6 e H7). Nesse ambiente, todo fluxo de tráfego gerado pelos servidores multimídia foi intermediado pelo *gateway* multimídia e, após processado por ele, encaminhado aos respectivos clientes da rede. Por sua

---

<sup>34</sup> iPerf em <https://iperf.fr>

<sup>35</sup> Wireshark em <https://www.wireshark.org>

<sup>36</sup> ntopng em <http://www.ntop.org/products/traffic-analysis/ntop>

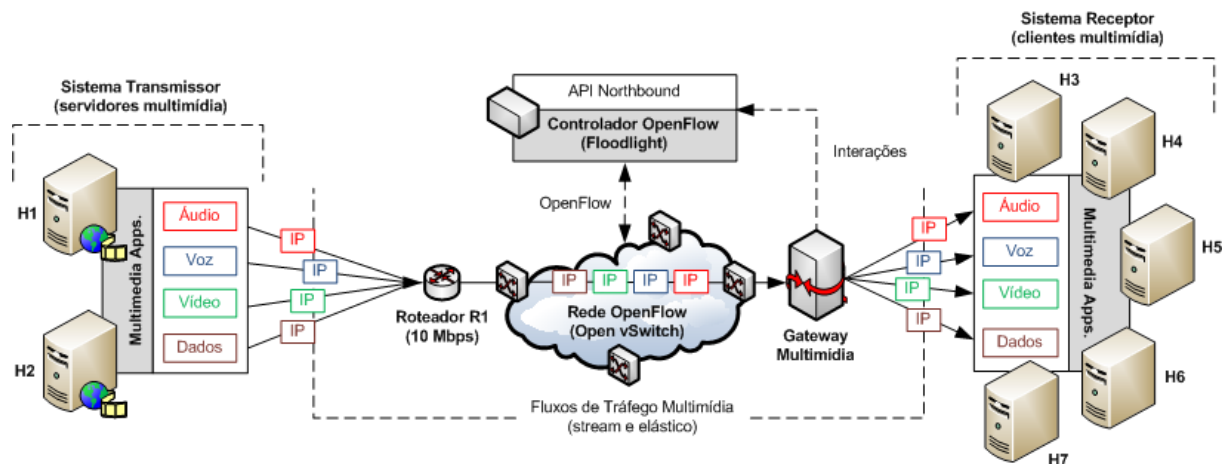
<sup>37</sup> MSU Quality Measurement Tool em [http://compression.ru/video/quality\\_measure/video\\_measurement\\_tool.html](http://compression.ru/video/quality_measure/video_measurement_tool.html)



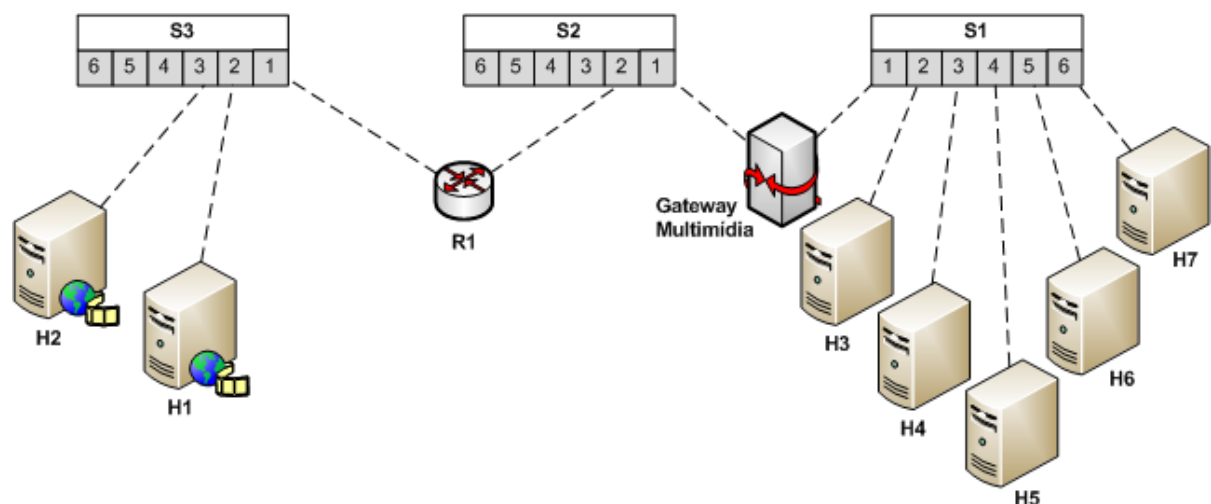
vez, as interações entre o *gateway* multimídia e o controlador SDN foram realizadas por meio das *northbound* REST APIs fornecidas pelo Floodlight.

Nesse contexto, como exemplo, a Figura 15 ilustra, de modo geral, o ambiente de emulação utilizado para a obtenção dos resultados experimentais; a Figura 16 ilustra a topologia experimental do ambiente de emulação; e, a Tabela 2 exibe as configurações de endereçamento IP utilizadas nos *hosts* desse ambiente em questão.

**Figura 15.** Ambiente de emulação utilizado para a obtenção de resultados experimentais: Sistemas transmissor e receptor dos fluxos de tráfego multimídia de áudio, de voz, de vídeo e de dados, roteador R1, controlador SDN e o *gateway* multimídia, ambos em um cenário SDN OpenFlow.



**Figura 16.** Topologia experimental do ambiente de emulação: *Switches* S1, S2 e S3 e suas interfaces de conexão com os *hosts* da rede.



**Tabela 2.** Configurações de endereçamento IP atribuídas aos *hosts* do ambiente de emulação.

<i>Host</i>	<b>Endereço(s) IP</b>
<b>Servidor Multimídia H1</b>	200.200.200.1/24
<b>Servidor Multimídia H2</b>	200.200.200.2/24
<b>Roteador R1</b>	200.200.200.254/24 188.188.188.2/24
<b>Gateway Multimídia</b>	188.188.188.1/24 172.18.0.254/16
<b>Cliente Multimídia H3</b>	172.18.0.1/16
<b>Cliente Multimídia H4</b>	172.18.0.2/16
<b>Cliente Multimídia H5</b>	172.18.0.3/16
<b>Cliente Multimídia H6</b>	172.18.0.4/16
<b>Cliente Multimídia H7</b>	172.18.0.5/16

Diante desse cenário, é possível observar que os *hosts* do ambiente de emulação estão interconectados à rede por meio de três *switches* distintos: *switches* S1, S2 e S3, ambos implementados por meio do *software switch* Open vSwitch. Nessa estrutura, o *switch* S1 é o responsável por viabilizar a interconexão dos clientes multimídia ao seu ambiente de rede local; o *switch* S3 é o responsável por viabilizar a interconexão dos servidores multimídia ao seu ambiente de rede local; e, por sua vez, o *switch* S2 é o responsável por viabilizar a interconexão entre esses dois ambientes de rede local, nesse caso, por meio da interconexão do *gateway* multimídia com o *switch* S1 e da interconexão do roteador R1 com o *switch* S3.

Nesse cenário, o *switch* de ingresso à rede dos servidores multimídia (*switch* S3) também é o responsável por atuar como adaptador de tráfego multimídia para o *gateway* multimídia. Nesse caso, é por meio dele que todos os fluxos de tráfego multimídia gerados pelo sistema transmissor são devidamente rotulados com seus respectivos identificadores de serviços multimídia.

Por sua vez, no nível IP, o ambiente de emulação foi logicamente organizado em três redes distintas (redes 172.18.0.0/16, 188.188.188.0/24 e 200.200.200.0/24), de modo a simular, por exemplo, um processo de distribuição de conteúdos multimídia em um ambiente Internet minimalista (não enfatizando, por exemplo, questões e particularidades voltadas à organização da rede em termos de Sistemas Autônomos, Pontos de Troca de Tráfego ou protocolos de roteamento inter-AS e intra-AS, dentre outros). Nessa estrutura, tanto o roteador

R1, quanto o *gateway* multimídia, atuam como *gateways* de acesso à rede para seus respectivos ambientes de redes locais. Nesse caso, ambos possuem duas configurações de endereçamento IP: uma configuração de endereçamento IP para viabilizar sua interoperação com seu ambiente de rede local (nesse caso, endereços IP 172.18.0.254/16 no *gateway* multimídia e 200.200.200.254/24 no roteador R1) e uma configuração de endereçamento IP para viabilizar suas interoperações entre si (nesse caso, endereços IP 188.188.188.1/24 no *gateway* multimídia e 188.188.188.2/24 no roteador R1).

Nesse contexto, de modo a exemplificar essa organização lógica de rede, a Figura 17 ilustra os saltos (*hops*) entre um dos clientes multimídia (*host* H7) e um dos servidores multimídia (*host* H2) no ambiente de emulação.

**Figura 17.** Saltos (*hops*) entre um dos clientes multimídia (*host* H7) e um dos servidores multimídia (*host* H2) no ambiente de emulação.

```
rfdiorio@h7:~$ traceroute h2.diorio.corp.br
traceroute to h2.diorio.corp.br (200.200.200.2), 30 hops max, 60 byte packets
 1 gateway.diorio.corp.br (172.18.0.254)  3.721 ms  2.402 ms  6.201 ms
 2 r1.diorio.corp.br (188.188.188.2)  15.446 ms  14.877 ms  14.598 ms
 3 h2.diorio.corp.br (200.200.200.2)  17.910 ms  17.836 ms  17.493 ms
```

Nesse exemplo, a partir do cliente multimídia H7, é possível observar que, no nível IP, o servidor multimídia H2 é acessível por meio do *gateway* multimídia (salto 1) e, em seguida, por meio do roteador R1 (salto 2), tal como descrito anteriormente quanto à organização lógica do ambiente de emulação.

Por sua vez, no ambiente de simulação, todo processo funcional do módulo preditor embasado no filtro de Kalman e em seu subfiltro também considerou os mesmos serviços e identificadores de serviços multimídia empregados no ambiente de emulação. Por outro lado, é importante destacar que a simulação não considerou eventuais cargas de dados (isto é, conteúdos multimídia emulados), bem como de particularidades no âmbito da rede e das aplicações quanto ao processo de comunicação multimídia, tais como quanto ao estabelecimento de conexões, ocorrências de atrasos, perdas de pacotes e retransmissões, dentre outros, enfatizando apenas no fornecimento dos identificadores de serviços multimídia para o módulo preditor embasado no filtro de Kalman e em seu subfiltro.

## 4.2 – Resultados e Discussão

Esta Seção discorre sobre os resultados obtidos por meio dos ambientes de emulação e simulação apresentados na Seção anterior, com ênfase quanto à rotulação de fluxos de tráfego multimídia, à identificação e encaminhamento diferenciado de fluxos de tráfego multimídia, às características de tráfego, tempos de transmissão, variações do atraso e perda de pacotes, ao encaminhamento por fluxos com suporte à QoS, ao cache de conteúdos multimídia e à predição de conteúdos multimídia, bem como de uma discussão quanto a tais resultados, de modo geral.

### 4.2.1 – Rotulação de Fluxos de Tráfego Multimídia

A Tabela 3 apresenta os fluxos de tráfego multimídia e seus respectivos identificadores de serviços multimídia (via rótulos DSCP) experimentados no ambiente de emulação. Em tal ambiente, esse processo de rotulação de fluxos de tráfego multimídia é realizado por meio do adaptador de tráfego multimídia.

**Tabela 3.** Fluxos de tráfego multimídia e seus respectivos identificadores de serviços multimídia, por meio de rótulos DSCP, experimentados no ambiente de emulação.

Fluxo de Tráfego Multimídia	Identificador (Rótulo DSCP)
Dados (via gerador de tráfego multimídia, canal 1)	000001 (0x01)
Dados (via gerador de tráfego multimídia, canal 2)	000010 (0x02)
Dados (via serviço de hospedagem <i>web</i> sobre HTTP)	000011 (0x03)
Dados (via serviço de transferência de arquivos sobre FTP)	000100 (0x04)
Áudio (via gerador de tráfego multimídia, canal 1)	010001 (0x11)
Áudio (via gerador de tráfego multimídia, canal 2)	010010 (0x12)
Áudio (via serviço de <i>streaming</i> de áudio sobre RTP e RTSP)	010011 (0x13)
Voz (via gerador de tráfego multimídia, canal 1)	100001 (0x21)
Voz (via gerador de tráfego multimídia, canal 2)	100010 (0x22)
Vídeo (via gerador de tráfego multimídia, canal 1)	110001 (0x31)
Vídeo (via gerador de tráfego multimídia, canal 2)	110010 (0x32)
Vídeo (via serviço de <i>streaming</i> de vídeo sobre RTP e RTSP)	110011 (0x33)

Diante desse cenário, regras de fluxos específicas para cada fluxo de tráfego multimídia em questão foram definidas junto ao controlador OpenFlow para, por sua vez, serem aplicadas ao adaptador de tráfego multimídia. Para tal, diferentes *match fields* do OpenFlow foram empregados na identificação de cada fluxo de tráfego multimídia, tais como por meio dos endereços IP dos servidores multimídia, das portas de comunicação dos serviços multimídia e dos endereços IP dos clientes multimídia. Como exemplo, a Figura 18, que é pertinente a listagem de regras de fluxos junto ao controlador OpenFlow, ilustra as definições empregadas para viabilizar a rotulação de dois fluxos experimentados no ambiente de emulação, nesse caso, fluxos de dados pertinentes ao serviço de hospedagem *web* (via HTTP) e fluxos de voz pertinentes ao canal 1 do gerador de tráfego multimídia.

**Figura 18.** Exemplo de definições para a rotulação de dois fluxos experimentados no ambiente de emulação: fluxos de dados pertinentes ao serviço de hospedagem *web* (via HTTP) e fluxos de voz pertinentes ao canal 1 do gerador de tráfego multimídia.

```

"data-http-1": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->3,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x6",
    "ipv4_dst": "172.18.0.2",
    "ipv4_src": "200.200.200.1",
    "tcp_src": "80"
  },
  [...]
  "version": "OF_13"
}

"voice-1": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->33,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x11",
    "ipv4_dst": "172.18.0.4",
    "ipv4_src": "200.200.200.2",
    "udp_dst": "8001"
  },
  [...]
  "version": "OF_13"
}

```

Nesse exemplo, quanto às informações ilustradas na Figura 18, os nomes “*data-http-1*” e “*voice-1*” referem-se aos nomes utilizados para identificar as duas regras de fluxos em questão junto ao controlador OpenFlow. Nesse caso, “*data-http-1*” refere-se aos fluxos de dados pertinentes ao serviço de hospedagem *web* (via HTTP) e “*voice-1*” refere-se aos fluxos de voz (canal 1) pertinentes ao gerador de tráfego multimídia. Por sua vez, em ambas as regras de fluxos, instruções distintas de rotulação são definidas de acordo com seus respectivos *match fields* OpenFlow. Nesse caso, na regra de fluxo “*data-http-1*”, a instrução é rotular os fluxos de tráfego utilizando o identificador DSCP 0x03, equivalente ao decimal 3 (“*actions*”: “*set\_field=ip\_dscp->3,output=normal*”). Para tal, essa regra é aplicada aos fluxos gerados pelo servidor multimídia H1 (“*ipv4\_src*”: “*200.200.200.1*”), em sua porta de origem 80/TCP (“*tcp\_src*”: “*80*”) e destinados ao cliente multimídia H4 (“*ipv4\_dst*”: “*172.18.0.2*”), nesse caso, utilizando informações dos protocolos IPv4 (“*eth\_type*”: “*0x800*”) e TCP (“*ip\_proto*”: “*0x6*”). Por sua vez, na regra de fluxo “*voice-1*”, a instrução é rotular os fluxos de tráfego utilizando o identificador DSCP 0x21 (decimal 33). Para tal, essa regra é aplicada aos fluxos gerados pelo servidor multimídia H2 e destinados ao cliente multimídia H6 em sua porta de destino 8001/UDP (“*udp\_dst*”: “*8001*”), nesse caso, utilizando informações dos protocolos IPv4 e UDP (“*ip\_proto*”: “*0x11*”). Em ambas as regras, a versão empregada do protocolo OpenFlow é a 1.3 (“*version*”: “*OF\_13*”).

Em tais exemplos, é importante destacar que apenas as informações pertinentes aos nomes das regras de fluxos e suas respectivas definições em termos de instruções (ações) e *match fields*, bem como quanto a versão empregada do protocolo OpenFlow, são ilustradas em tal figura. As demais informações, tidas como não relevantes para o exemplo em questão, foram substituídas e ilustradas pela *string* “[...]”.

Nesse contexto, nos mesmos moldes das regras de fluxos ilustradas por meio da Figura 18, regras de fluxos para rotulação de todos os fluxos experimentados no ambiente de emulação (Tabela 3) foram definidas junto ao controlador OpenFlow para, por sua vez, serem aplicadas ao adaptador de tráfego multimídia. Caso o leitor se interesse, todas essas regras de fluxos estão descritas no Apêndice A desta Tese (Seção A.1).

Diante desse cenário, todos os fluxos de tráfego multimídia experimentados no ambiente de emulação foram devidamente rotulados com seus respectivos identificadores DSCP. Como exemplo, as Figuras 19 e 20 ilustram, respectivamente, os identificadores DSCP atribuídos aos fluxos de dados pertinentes ao serviço de hospedagem *web* via HTTP e aos fluxos de voz pertinentes ao canal 1 do gerador de tráfego multimídia.

**Figura 19.** Exemplo de fluxos de tráfego multimídia de dados sobre HTTP e seus respectivos identificadores DSCP no ambiente de emulação.

Filter: **http** Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
4	0.063247000	172.18.0.2	200.200.200.1	HTTP	339	GET / HTTP/1.1
5	0.091950000	200.200.200.1	172.18.0.2	HTTP	763	HTTP/1.1 200 OK (text/html)
7	3.593764000	172.18.0.2	200.200.200.1	HTTP	381	GET /sample.zip HTTP/1.1
8664	15.970806000	200.200.200.1	172.18.0.2	HTTP	1007	HTTP/1.1 200 OK (application/x-zip-compressed)
8668	29.463525000	172.18.0.2	200.200.200.1	HTTP	339	GET / HTTP/1.1
8669	29.487076000	200.200.200.1	172.18.0.2	HTTP	763	HTTP/1.1 200 OK (text/html)

Frame 8664: 1007 bytes on wire (8056 bits), 1007 bytes captured (8056 bits)

Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:f5:fe:76 (08:00:27:f5:fe:76)

Internet Protocol Version 4, Src: 200.200.200.1 (200.200.200.1), Dst: 172.18.0.2 (172.18.0.2)

Version: 4  
Header length: 20 bytes

Differentiated Services Field: 0x0c (DSCP 0x03: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

**0000 11.. = Differentiated Services Codepoint: Unknown (0x03)**

... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)

Total Length: 993  
Identification: 0x4a3d (19005)

Flags: 0x02 (Don't Fragment)  
Fragment offset: 0  
Time to live: 126  
Protocol: TCP (6)

Header checksum: 0x71ef [correct]  
Source: 200.200.200.1 (200.200.200.1)  
Destination: 172.18.0.2 (172.18.0.2)

Transmission Control Protocol, Src Port: 80 (80), Dst Port: 1035 (1035), Seq: 8366394, Ack: 613, Len: 953

[5820 Reassembled TCP Segments (8366637 bytes): #8(1460), #9(1460), #10(1460), #12(1460), #13(1460), #15(1460), #16(1460)]

**Figura 20.** Exemplo de fluxos de tráfego multimídia de voz pertinentes ao canal 1 do gerador de tráfego multimídia e seus respectivos identificadores DSCP no ambiente de emulação.

Filter: **udp.port==8001** Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
9	40.059172	200.200.200.2	172.18.0.4	UDP	95	Source port: 44458 Destination port: 8001
12	40.108844	200.200.200.2	172.18.0.4	UDP	95	Source port: 45702 Destination port: 8001
13	40.136749	200.200.200.2	172.18.0.4	UDP	95	Source port: 34327 Destination port: 8001
26	40.229124	200.200.200.2	172.18.0.4	UDP	95	Source port: 55342 Destination port: 8001
36	40.273228	200.200.200.2	172.18.0.4	UDP	95	Source port: 59805 Destination port: 8001
50	40.323014	200.200.200.2	172.18.0.4	UDP	95	Source port: 43190 Destination port: 8001

Frame 12: 95 bytes on wire (760 bits), 95 bytes captured (760 bits)

Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:85:dc:eb (08:00:27:85:dc:eb)

Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.4 (172.18.0.4)

Version: 4  
Header length: 20 bytes

Differentiated Services Field: 0x84 (DSCP 0x21: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

**1000 01.. = Differentiated Services Codepoint: Unknown (0x21)**

... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)

Total Length: 81  
Identification: 0x447d (17533)

Flags: 0x02 (Don't Fragment)  
Fragment offset: 0  
Time to live: 62  
Protocol: UDP (17)

Header checksum: 0xbab9 [correct]  
Source: 200.200.200.2 (200.200.200.2)  
Destination: 172.18.0.4 (172.18.0.4)

User Datagram Protocol, Src Port: 45702 (45702), Dst Port: 8001 (8001)

Data (53 bytes)

#### 4.2.2 – Identificação e Encaminhamento Diferenciado de Fluxos de Tráfego Multimídia

Para avaliar a identificação e encaminhamento diferenciado de fluxos de tráfego multimídia na rede, foram utilizados, em especial, os fluxos multimídia de dados, de áudio, de voz e de vídeo, gerados por meio do gerador de tráfego multimídia, no ambiente de emulação.

Para tal, as aplicações transmissoras e receptoras do gerador de tráfego multimídia, em execução no servidor multimídia H2 (aplicação transmissora) e nos clientes multimídia H6 e H7 (aplicações receptoras), foram configuradas de modo a, por si próprias, não serem capazes de estabelecer um processo de comunicação na rede. Nesse caso, na aplicação transmissora do gerador de tráfego multimídia, o endereço IP dos clientes da rede (originalmente, pertinentes aos *hosts* H6 e H7) foi alterado para o endereço IP do *gateway* multimídia. Além disso, as portas de comunicação utilizadas para a transmissão dos conteúdos multimídia de dados, de áudio, de voz e de vídeo também receberam distintas configurações em ambas as aplicações, de modo que as portas empregadas na transmissão de ambos os conteúdos pela aplicação transmissora não são as mesmas empregadas na recepção dos mesmos pelas aplicações receptoras. De modo a contribuir com a explanação de tal cenário, a Tabela 4 sintetiza tais configurações junto ao ambiente de emulação.

**Tabela 4.** Síntese de configurações realizadas no âmbito das aplicações transmissoras e receptoras do gerador de tráfego multimídia quanto aos fluxos de dados, de áudio, de voz e de vídeo experimentados no ambiente de emulação.

Fluxo de tráfego multimídia de	Identificador (rótulo DSCP)	Fluxos transmitidos para		Fluxos esperados por	
		IP	Porta	IP	Porta
Dados (canal 1)	0x01	188.188.188.1	6111/TCP	172.18.0.4	6001/TCP
Dados (canal 2)	0x02	188.188.188.1	6112/TCP	172.18.0.5	6002/TCP
Áudio (canal 1)	0x11	188.188.188.1	7111/UDP	172.18.0.4	7001/UDP
Áudio (canal 2)	0x12	188.188.188.1	7112/UDP	172.18.0.5	7002/UDP
Voz (canal 1)	0x21	188.188.188.1	8111/UDP	172.18.0.4	8001/UDP
Voz (canal 2)	0x22	188.188.188.1	8112/UDP	172.18.0.5	8002/UDP
Vídeo (canal 1)	0x31	188.188.188.1	9111/UDP	172.18.0.4	9001/UDP
Vídeo (canal 2)	0x32	188.188.188.1	9112/UDP	172.18.0.5	9002/UDP



Diante desse cenário, com base nos identificadores DSCP pertinentes a cada fluxo de tráfego em questão, cabe ao *gateway* multimídia identificá-los e encaminhá-los corretamente aos seus respectivos clientes multimídia na rede. Para tal, deve fornecer um mecanismo de encaminhamento diferenciado para tais fluxos de tráfego multimídia, considerando apenas seus identificadores DSCP e com definições de encaminhamento considerando modificações nos níveis de rede (alterações nos endereços IP) e de transporte (alterações nas portas de comunicação) de seus destinatários.

Nesse contexto, regras de fluxos específicas para cada fluxo de tráfego multimídia em questão foram definidas junto ao controlador OpenFlow para, por sua vez, serem aplicadas ao *gateway* multimídia no ambiente de emulação. Em linhas gerais, essas regras empregaram como *match fields* do OpenFlow os identificadores DSCP pertinentes aos fluxos de tráfego multimídia que terão tratamento diferenciado na rede, bem como de referências aos protocolos IPv4 e TCP/UDP pelas modificações nos níveis de rede e de transporte no âmbito de tais encaminhamentos, de modo geral.

Como exemplo, a Figura 21, que é pertinente a listagem de regras de fluxos junto ao controlador OpenFlow, ilustra as definições empregadas para viabilizar o encaminhamento diferenciado de dois fluxos experimentados no ambiente de emulação, nesse caso, fluxos de dados pertinentes ao canal 1 do gerador de tráfego multimídia e fluxos de voz pertinentes ao canal 2 do gerador de tráfego multimídia.

Em tal exemplo, a regra de fluxo “*redir-data-1*” é pertinente ao encaminhamento diferenciado para os fluxos de dados no canal 1 e a regra de fluxo “*redir-voice-2*” é pertinente ao encaminhamento diferenciado para os fluxos de voz no canal 2, ambos quanto aos fluxos gerados por meio do gerador de tráfego multimídia. Por sua vez, em ambas as regras de fluxos, instruções distintas de encaminhamento são definidas de acordo com seus respectivos *match fields* OpenFlow. Nesse caso, na regra de fluxo “*redir-data-1*”, a instrução é direcionar os fluxos em questão para o cliente multimídia H6 na porta 6001/TCP (“*actions*”: “*set\_field=ipv4\_dst->172.18.0.4,set\_field=tcp\_dst->6001,output=normal*”). Para tal, essa regra é aplicada aos fluxos IPv4 (“*eth\_type*”: “*0x800*”) com identificador DSCP 0x01 (“*ip\_dscp*”: “*0x1*”) e que utilizam o protocolo TCP (“*ip\_proto*”: “*0x6*”) como protocolo de transporte. Por sua vez, na regra de fluxo “*redir-voice-2*”, a instrução é direcionar os fluxos em questão para o cliente multimídia H7 na porta 8002/UDP. Para tal, essa regra é aplicada aos fluxos IPv4 com identificador DSCP 0x22 e que utilizam o protocolo UDP (“*ip\_proto*”:

"0x11") como protocolo de transporte. Em ambas as regras, a versão empregada do protocolo OpenFlow é a 1.3 ("version": "OF\_13").

**Figura 21.** Exemplo de definições para o encaminhamento diferenciado de dois fluxos experimentados no ambiente de emulação: fluxos de dados pertinentes ao canal 1 do gerador de tráfego multimídia e fluxos de voz pertinentes ao canal 2 do gerador de tráfego multimídia.

```

    "redir-data-1": {
      [...]
      "instructions": {
        "instruction_apply_actions": {
          "actions": "set_field=ipv4_dst->172.18.0.4,
set_field=tcp_dst->6001,output=normal"
        }
      },
      "match": {
        "eth_type": "0x800",
        "ip_dscp": "0x1",
        "ip_proto": "0x6"
      },
      [...]
      "version": "OF_13"
    }

    "redir-voice-2": {
      [...]
      "instructions": {
        "instruction_apply_actions": {
          "actions": "set_field=ipv4_dst->172.18.0.5,
set_field=udp_dst->8002,output=normal"
        }
      },
      "match": {
        "eth_type": "0x800",
        "ip_dscp": "0x22",
        "ip_proto": "0x11"
      },
      [...]
      "version": "OF_13"
    }
  
```

Em tal exemplo, assim como realizado na Seção anterior, é importante destacar que apenas as informações pertinentes aos nomes das regras de fluxos e suas respectivas definições em termos de instruções (ações) e *match fields*, bem como quanto a versão empregada do protocolo OpenFlow, são ilustradas em tal figura. As demais informações, tidas como não relevantes para o exemplo em questão, foram substituídas e ilustradas pela *string* “[...]”.

Nesse cenário, de modo complementar a tais definições, pelo fato dos fluxos de dados empregarem o protocolo TCP como protocolo de transporte (nesse caso, com questões envolvendo o estabelecimento, a manutenção e o encerramento de conexões no nível de transporte), regras de fluxos complementares devem ser empregadas de modo a direcionar as respostas dos clientes multimídia H6 e H7 para o servidor multimídia H2. Nesse caso, pelo fato do servidor multimídia ter o *gateway* multimídia como cliente multimídia, essas respostas devem ser direcionadas a ele (servidor multimídia) como originadas pelo *gateway* multimídia. Por sua vez, essas respostas também devem ser direcionadas considerando as portas de comunicação empregadas por sua aplicação transmissora.

Dessa forma, os fluxos TCP originados pelo cliente multimídia H6 em sua porta de origem 6001/TCP (dados, canal 1), como resposta aos fluxos transmitidos pelo servidor multimídia H2 ao *gateway* multimídia em sua porta de destino 6111/TCP, devem ser direcionados ao servidor multimídia H2 como respostas do *gateway* multimídia em sua porta de origem 6111/TCP. Da mesma forma, os fluxos TCP originados pelo cliente multimídia H7 em sua porta de origem 6002/TCP (dados, canal 2), como resposta aos fluxos transmitidos pelo servidor multimídia H2 ao *gateway* multimídia em sua porta de destino 6112/TCP, devem ser direcionados ao servidor multimídia H2 como respostas do *gateway* multimídia em sua porta de origem 6112/TCP.

Nesse contexto, como exemplo, a Figura 22 ilustra as definições complementares empregadas para viabilizar o encaminhamento diferenciado dos fluxos de dados pertinentes ao canal 1 do gerador de tráfego multimídia no ambiente de emulação. É importante destacar que essas definições complementares foram necessárias somente para os fluxos de dados (canais 1 e 2) pertinentes ao gerador de tráfego multimídia no ambiente de emulação, nesse caso, pelo emprego do protocolo de transporte TCP na transmissão de tais fluxos de dados.

Por meio de tal ilustração, observe que o nome da regra de fluxo em questão é definido como *“resp-redirect-data-1”*. Em tal regra de fluxo, a instrução é direcionar as respostas pertinentes aos fluxos de dados no canal 1 do gerador de tráfego multimídia como originadas pelo *gateway* multimídia em sua porta 6111/TCP (*“actions”: “set\_field=ipv4\_src->188.188.188.1, set\_field=tcp\_src->6111,output=normal”*). Para tal, essa regra é aplicada aos fluxos IPv4 (*“eth\_type”: “0x800”*), que utilizam o protocolo TCP (*“ip\_proto”: “0x6”*) como protocolo de transporte, originados pelo cliente multimídia H6 (*“ipv4\_src”: “172.18.0.4”*) em sua porta 6001/TCP (*“tcp\_src”: “6001”*).

**Figura 22.** Exemplo de definições complementares para o encaminhamento diferenciado dos fluxos de dados pertinentes ao canal 1 do gerador de tráfego multimídia no ambiente de emulação.

```

    "resp-redirect-data-1": {
      [...]
      "instructions": {
        "instruction_apply_actions": {
          "actions": "set_field=ipv4_src->188.188.188.1,
set_field=tcp_src->6111,output=normal"
        }
      },
      "match": {
        "eth_type": "0x800",
        "ip_proto": "0x6",
        "ipv4_src": "172.18.0.4",
        "tcp_src": "6001"
      },
      [...]
      "version": "OF_13"
    }
  
```

Nesse contexto, nos mesmos moldes das regras de fluxos ilustradas por meio da Figura 21, bem como da Figura 22 enquanto definições complementares para os fluxos de dados, regras de fluxos para identificação e encaminhamento diferenciado de todos os fluxos experimentados no ambiente de emulação foram definidas junto ao controlador OpenFlow para, por sua vez, serem aplicadas ao *gateway* multimídia. Caso o leitor se interesse, todas essas regras de fluxos estão descritas no Apêndice A desta Tese (Seção A.2).

Diante desse cenário, durante a realização das avaliações experimentais, a identificação e o encaminhamento diferenciado de fluxos de tráfego multimídia, nesse caso, por meio de identificadores DSCP específicos, mostraram-se funcionais no ambiente de emulação. Em linhas gerais, houve um processo de comunicação entre as aplicações transmissoras e receptoras dos conteúdos multimídia de áudio, de voz, de vídeo e de dados no ambiente SDN. Por sua vez, com a alteração e/ou remoção de qualquer uma das regras de fluxos pertinentes aos conteúdos multimídia em questão, esse processo de comunicação era imediatamente percebido no ambiente de emulação, com o(s) fluxo(s) em questão deixando de ser encaminhado(s) ao(s) seu(s) respectivo(s) cliente(s) multimídia.

Como exemplo, as Figuras 23 e 24 ilustram, respectivamente, os fluxos multimídia de dados (canal 1) e de voz (canal 2) recebidos e encaminhados pelo *gateway* multimídia no ambiente de emulação.

**Figura 23.** Exemplo de recebimento (A) e encaminhamento diferenciado (B) de fluxos de tráfego multimídia de dados (canal 1) junto ao *gateway* multimídia no ambiente de emulação.

(A)

Filter: `frame.interface_id==1 and tcp.port==6111 or tcp.por` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
38	6.072394000	200.200.200.2	188.188.188.1	TCP	74	51295 > 6111 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SAC
40	6.080523000	188.188.188.1	200.200.200.2	TCP	74	6111 > 51295 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 M
42	6.088223000	200.200.200.2	188.188.188.1	TCP	66	51295 > 6111 [ACK] Seq=1 Ack=1 Win=29248 Len=0 TSval=
44	6.098135000	200.200.200.2	188.188.188.1	TCP	691	51295 > 6111 [PSH, ACK] Seq=1 Ack=1 Win=29248 Len=625
45	6.101744000	200.200.200.2	188.188.188.1	TCP	66	51295 > 6111 [FIN, ACK] Seq=626 Ack=1 Win=29248 Len=6
46	6.110812000	188.188.188.1	200.200.200.2	TCP	66	6111 > 51295 [ACK] Seq=1 Ack=626 Win=30272 Len=0 TSva

Frame 38: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 1  
 Ethernet II, Src: 08:00:27:d4:00:e9 (08:00:27:d4:00:e9), Dst: 08:00:27:2b:f2:57 (08:00:27:2b:f2:57)  
 Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 188.188.188.1 (188.188.188.1)  
 Version: 4  
 Header length: 20 bytes  
 Differentiated Services Field: 0x04 (DSCP 0x01: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0000 01.. = Differentiated Services Codepoint: Unknown (0x01)  
 .... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 60  
 Identification: 0xb215 (45589)  
 Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 63  
 Protocol: TCP (6)  
 Header checksum: 0x8019 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 188.188.188.1 (188.188.188.1)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 Transmission Control Protocol, Src Port: 51295 (51295), Dst Port: 6111 (6111), Seq: 0, Len: 0

(B)

Filter: `frame.interface_id==0 and tcp.port==6001 or tcp.por` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
9	6.076873000	200.200.200.2	172.18.0.4	TCP	74	51295 > 6001 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SAC
10	6.077912000	172.18.0.4	200.200.200.2	TCP	74	6001 > 51295 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 M
13	6.090502000	200.200.200.2	172.18.0.4	TCP	66	51295 > 6001 [ACK] Seq=1 Ack=1 Win=29248 Len=0 TSval=
15	6.098181000	200.200.200.2	172.18.0.4	TCP	691	[TCP segment of a reassembled PDU]
16	6.101793000	200.200.200.2	172.18.0.4	TCP	66	51295 > 6001 [FIN, ACK] Seq=626 Ack=1 Win=29248 Len=6
17	6.102808000	172.18.0.4	200.200.200.2	TCP	66	6001 > 51295 [ACK] Seq=1 Ack=626 Win=30272 Len=0 TSva

Frame 9: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0  
 Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:85:dc:eb (08:00:27:85:dc:eb)  
 Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.4 (172.18.0.4)  
 Version: 4  
 Header length: 20 bytes  
 Differentiated Services Field: 0x04 (DSCP 0x01: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0000 01.. = Differentiated Services Codepoint: Unknown (0x01)  
 .... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 60  
 Identification: 0xb215 (45589)  
 Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 62  
 Protocol: TCP (6)  
 Header checksum: 0x4dc1 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 172.18.0.4 (172.18.0.4)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 Transmission Control Protocol, Src Port: 51295 (51295), Dst Port: 6001 (6001), Seq: 0, Len: 0

**Figura 24.** Exemplo de recebimento (A) e encaminhamento diferenciado (B) de fluxos de tráfego multimídia de voz (canal 2) junto ao *gateway* multimídia no ambiente de emulação.

Filter: `interface_id==1 and udp.port==8111 or udp.port==811` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
349	7.905572000	200.200.200.2	188.188.188.1	UDP	95	Source port: 54209 Destination port: 8111
364	7.970690000	200.200.200.2	188.188.188.1	UDP	95	Source port: 38611 Destination port: 8112
365	7.970710000	200.200.200.2	188.188.188.1	UDP	95	Source port: 53972 Destination port: 8111
424	8.007602000	200.200.200.2	188.188.188.1	UDP	95	Source port: 60430 Destination port: 8112
426	8.013005000	200.200.200.2	188.188.188.1	UDP	95	Source port: 49724 Destination port: 8111
438	8.061388000	200.200.200.2	188.188.188.1	UDP	95	Source port: 39338 Destination port: 8111

Frame 364: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface 1  
 Ethernet II, Src: 08:00:27:d4:00:e9 (08:00:27:d4:00:e9), Dst: 08:00:27:2b:f2:57 (08:00:27:2b:f2:57)  
 Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 188.188.188.1 (188.188.188.1)  
 Version: 4  
 Header length: 20 bytes  
 Differentiated Services Field: 0x88 (DSCP 0x22: Assured Forwarding 41; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 1000 10.. = Differentiated Services Codepoint: Assured Forwarding 41 (0x22)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 81  
 Identification: 0x0233 (563)  
 Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 63  
 Protocol: UDP (17)  
 Header checksum: 0x2f58 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 188.188.188.1 (188.188.188.1)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 User Datagram Protocol, Src Port: 38611 (38611), Dst Port: 8112 (8112)

Filter: `frame.interface_id==0 and udp.port==8001 or udp.port==8002` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
315	7.911559000	200.200.200.2	172.18.0.4	UDP	95	Source port: 54209 Destination port: 8001
370	7.976994000	200.200.200.2	172.18.0.4	UDP	95	Source port: 53972 Destination port: 8001
371	7.977127000	200.200.200.2	172.18.0.5	UDP	95	Source port: 38611 Destination port: 8002
386	8.035240000	200.200.200.2	172.18.0.5	UDP	95	Source port: 60430 Destination port: 8002
387	8.035342000	200.200.200.2	172.18.0.4	UDP	95	Source port: 49724 Destination port: 8001
395	8.067567000	200.200.200.2	172.18.0.4	UDP	95	Source port: 39338 Destination port: 8001

Frame 371: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface 0  
 Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:c6:13:07 (08:00:27:c6:13:07)  
 Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.5 (172.18.0.5)  
 Version: 4  
 Header length: 20 bytes  
 Differentiated Services Field: 0x88 (DSCP 0x22: Assured Forwarding 41; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 1000 10.. = Differentiated Services Codepoint: Assured Forwarding 41 (0x22)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 81  
 Identification: 0x0233 (563)  
 Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 62  
 Protocol: UDP (17)  
 Header checksum: 0xfcfe [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 172.18.0.5 (172.18.0.5)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 User Datagram Protocol, Src Port: 38611 (38611), Dst Port: 8002 (8002)

Em tais ilustrações, observe que os fluxos recebidos pelo *gateway* multimídia possuem seu endereço IP como destinatário, bem como as portas de comunicação especificadas anteriormente junto à aplicação transmissora do gerador de tráfego multimídia. Por outro lado, os fluxos encaminhados pelo *gateway* multimídia possuem os endereços IP dos clientes multimídia H6 e H7 como destinatários, bem como suas respectivas portas de escuta no âmbito da aplicação receptora do gerador de tráfego multimídia.

De modo complementar, por meio da Figura 25, que ilustra o recebimento dos fluxos de tráfego multimídia de dados, de áudio, de voz e de vídeo pelo cliente multimídia H6, ambos gerados por meio do gerador de tráfego multimídia no canal 1, observe que todos os fluxos recebidos possuem o endereço IP do servidor multimídia H2 enquanto sistema transmissor, de modo que, em linhas gerais, o *gateway* multimídia foi tido como “transparente” durante esse processo de encaminhamento diferenciado de fluxos de tráfego multimídia no ambiente de emulação.

**Figura 25.** Exemplo de recebimento de fluxos multimídia de dados, de áudio, de voz e de vídeo pelo cliente multimídia H6 no ambiente de emulação, ambos gerados por meio do gerador de tráfego multimídia no canal 1.

The screenshot shows the Wireshark interface with a list of captured packets. A red dashed box highlights packets 3 through 8. Red arrows point to the first, fourth, and eighth packets in the list. The detailed view of the selected packet (No. 8) is shown below, with red arrows pointing to the Ethernet II, Internet Protocol, and Transmission Control Protocol layers.

No.	Time	Source	Destination	Protocol	Length	Info
3	6.076811	200.200.200.2	172.18.0.4	TCP	74	51295 > 6001 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SAC...
4	6.076842	172.18.0.4	200.200.200.2	TCP	74	6001 > 51295 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MS...
5	6.080170	200.200.200.2	172.18.0.4	UDP	95	Source port: 60353 Destination port: 8001
6	6.085768	200.200.200.2	172.18.0.4	UDP	135	Source port: 34542 Destination port: 9001
7	6.090383	200.200.200.2	172.18.0.4	TCP	66	51295 > 6001 [ACK] Seq=1 Ack=1 Win=29248 Len=0 TSval=...
8	6.093761	200.200.200.2	172.18.0.4	UDP	105	Source port: 49369 Destination port: 7001

Frame 3: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)  
 Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:85:dc:eb (08:00:27:85:dc:eb)  
 Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.4 (172.18.0.4)  
 Version: 4  
 Header length: 20 bytes  
 Differentiated Services Field: 0x04 (DSCP 0x01: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0000 01.. = Differentiated Services Codepoint: Unknown (0x01)  
 .... 0000 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 60  
 Identification: 0xb215 (45589)  
 Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 62  
 Protocol: TCP (6)  
 Header checksum: 0x4dc1 [correct]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 172.18.0.4 (172.18.0.4)  
 Transmission Control Protocol, Src Port: 51295 (51295), Dst Port: 6001 (6001), Seq: 0, Len: 0

### 4.2.3 – Características de Tráfego, Tempos de Transmissão, Variações do Atraso e Perda de Pacotes

Para avaliar questões envolvendo características de tráfego, tempos de transmissão, variações do atraso (*jitter*) e perda de pacotes, foram utilizados o gerador de tráfego multimídia e a ferramenta iPerf, ambas no ambiente de emulação.

Nesse contexto, quanto às soluções empregadas para tais avaliações, de modo complementar à geração dos fluxos multimídia de dados, de áudio, de voz e de vídeo, o gerador de tráfego multimídia também possibilita a realização de observações sobre as características de tais fluxos de tráfego junto à aplicação receptora de seus conteúdos. Em linhas gerais, por meio de um arquivo de *log* fornecido por tal aplicação, é possível observar se a frequência e os tempos de cada pacote recebido pelo sistema receptor estão de acordo com a distribuição empregada na geração dos mesmos pelo sistema transmissor. Por outro lado, por não fornecer informações sobre tempos de transmissão, variações do atraso e pacotes perdidos, a ferramenta iPerf foi empregada para tal. Dessa forma, por meio de tais soluções, é possível observar se há eventuais alterações nas características dos fluxos de tráfego gerados por meio do gerador de tráfego multimídia, bem como quanto aos tempos de transmissão, variações do atraso e percentual de datagramas perdidos para os fluxos de tráfego gerados por meio do iPerf, ambos durante o processo de encaminhamento realizado pelo *gateway* multimídia.

Diante desse cenário, quatro experimentações foram realizadas no ambiente de emulação:

- (I) Transmissões de áudio, de voz e de vídeo, com fluxos de tráfego gerados por meio do gerador de tráfego multimídia, sem o intermédio do *gateway* multimídia;
- (II) Transmissões de áudio, de voz e de vídeo, com fluxos de tráfego gerados por meio do gerador de tráfego multimídia, com o intermédio do *gateway* multimídia;
- (III) Transmissões de fluxos de tráfego aleatórios, gerados por meio do iPerf, sobre o protocolo UDP, sem o intermédio do *gateway* multimídia;



(IV) Transmissões de fluxos de tráfego aleatórios, gerados por meio do iPerf, sobre o protocolo UDP, com o intermédio do *gateway* multimídia.

Nesse cenário, quanto aos fluxos de áudio, de voz e de vídeo, gerados por meio do gerador de tráfego multimídia, transmissões de 30 minutos, de 1 hora, de 3 horas e de 6 horas foram consideradas para cada experimentação, ambas realizadas entre o servidor multimídia H2 e os clientes multimídia H6 (sem o intermédio do *gateway* multimídia) e H7 (com o intermédio do *gateway* multimídia). Nesse caso, os tempos de transmissão empregados em tais experimentações foram escolhidos aleatoriamente, inclusive, com menores e maiores valores quanto a ambos em outras experimentações realizadas no decorrer do desenvolvimento desta Tese. Por sua vez, ambos os fluxos de tráfego em questão foram modelados por meio da função de distribuição exponencial.

Nesse contexto, quanto aos fluxos de tráfego aleatórios gerados por meio do iPerf e sobre o protocolo UDP, para transmissões com durações aproximadas às empregadas no gerador de tráfego multimídia, considerou-se fluxos de 225 MB (transmissões de 30 minutos), 450 MB (transmissões de 1 hora), 1350 MB (transmissões de 3 horas) e 2700 MB (transmissões de 6 horas). Em tais transmissões, a escolha quanto ao tamanho dos fluxos teve como base o tempo de 8 segundos para cada 1 MB transmitido junto ao iPerf em sua configuração padrão. Nesse caso, para cada fluxo de tráfego em questão, 5 transmissões foram consideradas (transmissões de T1 à T5), ambas realizadas entre o servidor multimídia H1 e os clientes multimídia H3 (com o intermédio do *gateway* multimídia) e H4 (sem o intermédio do *gateway* multimídia).

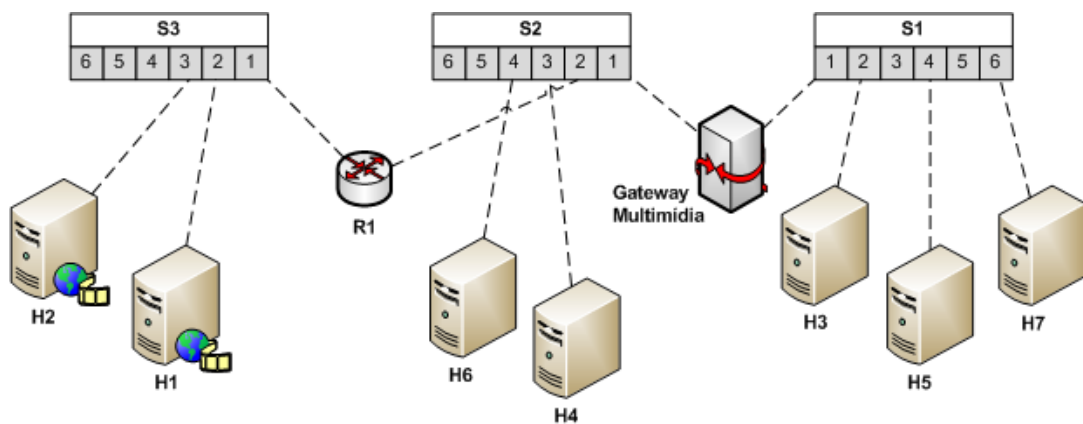
Em tais experimentações, é importante destacar que apenas uma solução de *software* foi empregada por vez (gerador de tráfego multimídia ou iPerf). Por sua vez, os fluxos de tráfego gerados por meio de ambas, nesse caso a partir dos servidores multimídia H1 (iPerf) e H2 (gerador de tráfego multimídia), foram transmitidos simultaneamente para os clientes multimídia H3 e H4 (iPerf) e H6 e H7 (gerador de tráfego multimídia).

Diante desse cenário, a topologia experimental do ambiente de emulação, ilustrada anteriormente na Figura 16, foi alterada momentaneamente para a realização das experimentações descritas nesta Seção. Nesse caso, os clientes multimídia H4 e H6, utilizados nas transmissões diretas (sem o intermédio do *gateway* multimídia) com os servidores multimídia H1 e H2, respectivamente, foram movidos do *switch* S1 para o *switch* S2, sendo interconectados, respectivamente, nas interfaces 3 e 4 de tal *switch* S2. Além disso, no nível

de rede, os clientes multimídia em questão também tiveram suas configurações de endereçamento IP alteradas. Nesse caso, o cliente multimídia H4 teve seu endereço IP alterado para 188.188.188.3/24 e, por sua vez, o cliente multimídia H6 teve seu endereço IP alterado para 188.188.188.4/24. Em linhas gerais, essas alterações foram necessárias para possibilitar que esses clientes multimídia, em específico, pudessem se comunicar diretamente com os servidores multimídia da rede, nesse caso, sem o intermédio do *gateway* multimídia.

Nesse contexto, como exemplo, a Figura 26 ilustra a topologia experimental do ambiente de emulação para a realização das experimentações descritas anteriormente. De modo complementar, as Figuras 27 e 28, ilustram, respectivamente, os saltos (*hops*) entre os clientes multimídia H6 e H7 e o servidor multimídia H2 (utilizados nas experimentações com o gerador de tráfego multimídia) e entre os clientes multimídia H3 e H4 e o servidor multimídia H1 (utilizados nas experimentações com o iPerf).

**Figura 26.** Topologia experimental para as avaliações sobre as características de tráfego, tempos de transmissão, variações do atraso e perda de pacotes no ambiente de emulação.



**Figura 27.** Saltos (*hops*) entre o cliente multimídia H6 e o servidor multimídia H2 (A) e entre o cliente multimídia H7 e o servidor multimídia H2 (B), ambos utilizados nas experimentações com o gerador de tráfego multimídia.

```
rfdiorio@h6:~$ traceroute h2.diorio.corp.br
traceroute to h2.diorio.corp.br (200.200.200.2), 30 hops max, 60 byte packets
 1  r1.diorio.corp.br (188.188.188.2)  23.473 ms  25.368 ms  25.184 ms
 2  h2.diorio.corp.br (200.200.200.2)  24.983 ms  24.744 ms  24.313 ms
```

(A)

```
rfdiorio@h7:~$ traceroute h2.diorio.corp.br
traceroute to h2.diorio.corp.br (200.200.200.2), 30 hops max, 60 byte packets
 1  gateway.diorio.corp.br (172.18.0.254)  3.721 ms  2.402 ms  6.201 ms
 2  r1.diorio.corp.br (188.188.188.2)  15.446 ms  14.877 ms  14.598 ms
 3  h2.diorio.corp.br (200.200.200.2)  17.910 ms  17.836 ms  17.493 ms
```

(B)

**Figura 28.** Saltos (*hops*) entre o cliente multimídia H4 e o servidor multimídia H1 (A) e entre o cliente multimídia H3 e o servidor multimídia H1 (B), ambos utilizados nas experimentações com o iPerf.

```
C:\Documents and Settings\aluno>tracert h1.diorio.corp.br
Tracing route to h1.diorio.corp.br [200.200.200.1]
over a maximum of 30 hops:
  1    1 ms    <1 ms    <1 ms    r1.diorio.corp.br [188.188.188.2]
  2    3 ms    3 ms    3 ms    h1.diorio.corp.br [200.200.200.1]
```

(A)

```
C:\Documents and Settings\aluno>tracert h1.diorio.corp.br
Tracing route to h1.diorio.corp.br [200.200.200.1]
over a maximum of 30 hops:
  1    <1 ms    <1 ms    <1 ms    gateway.diorio.corp.br [172.18.0.254]
  2    3 ms    1 ms    2 ms    r1.diorio.corp.br [188.188.188.2]
  3    1 ms    <1 ms    <1 ms    h1.diorio.corp.br [200.200.200.1]
```

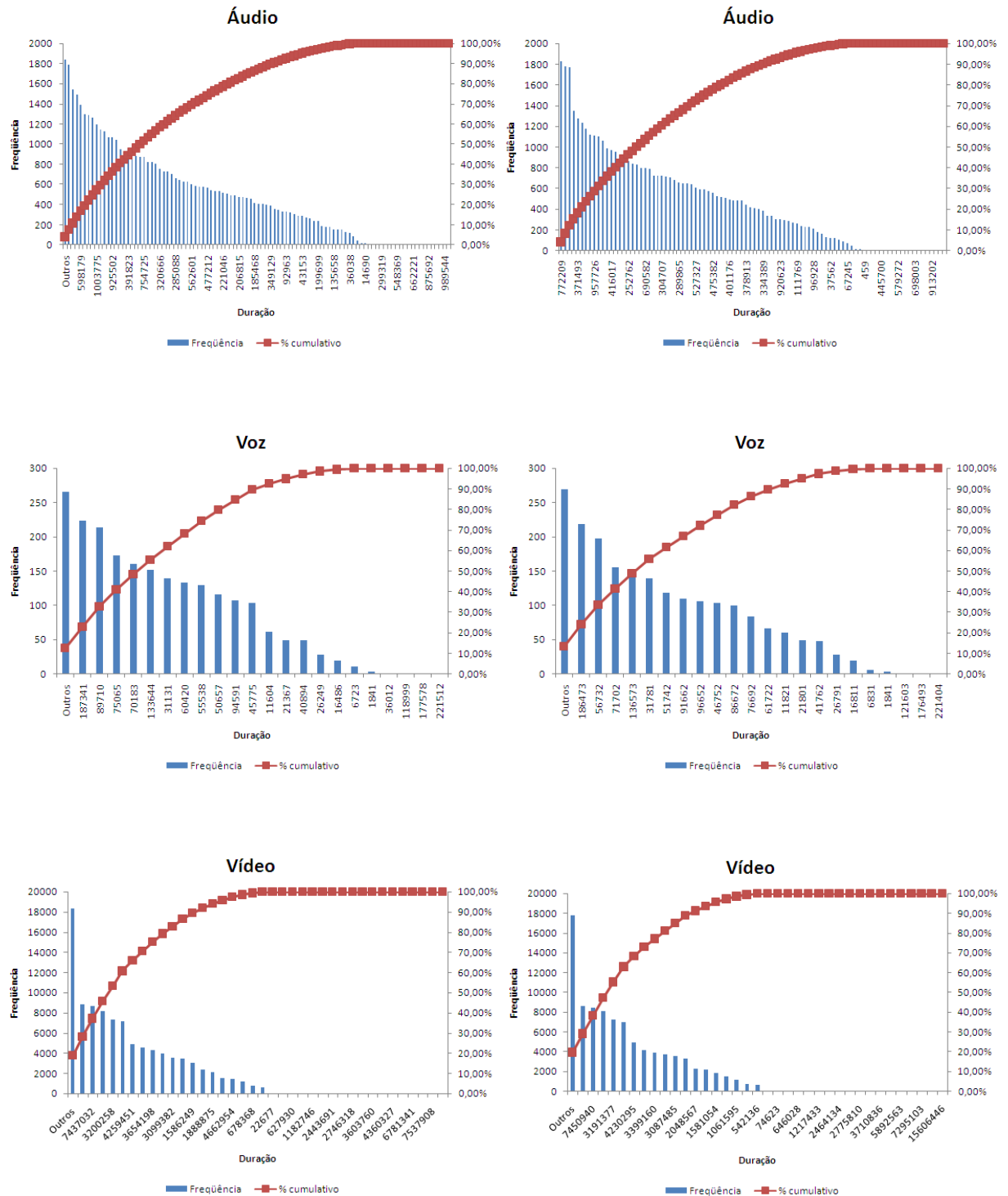
(B)

Nesse contexto, por meio das Figuras 27 e 28, é possível observar que os clientes multimídia H3 e H7 alcançam, respectivamente e no nível IP, os servidores multimídia H1 e H2 por meio do *gateway* multimídia (salto 1) e, em seguida, por meio do roteador R1 (salto 2). Por sua vez, os clientes multimídia H4 e H6 alcançam tais servidores diretamente por meio do roteador R1 (salto 1), nesse caso, sem o intermédio do *gateway* multimídia.

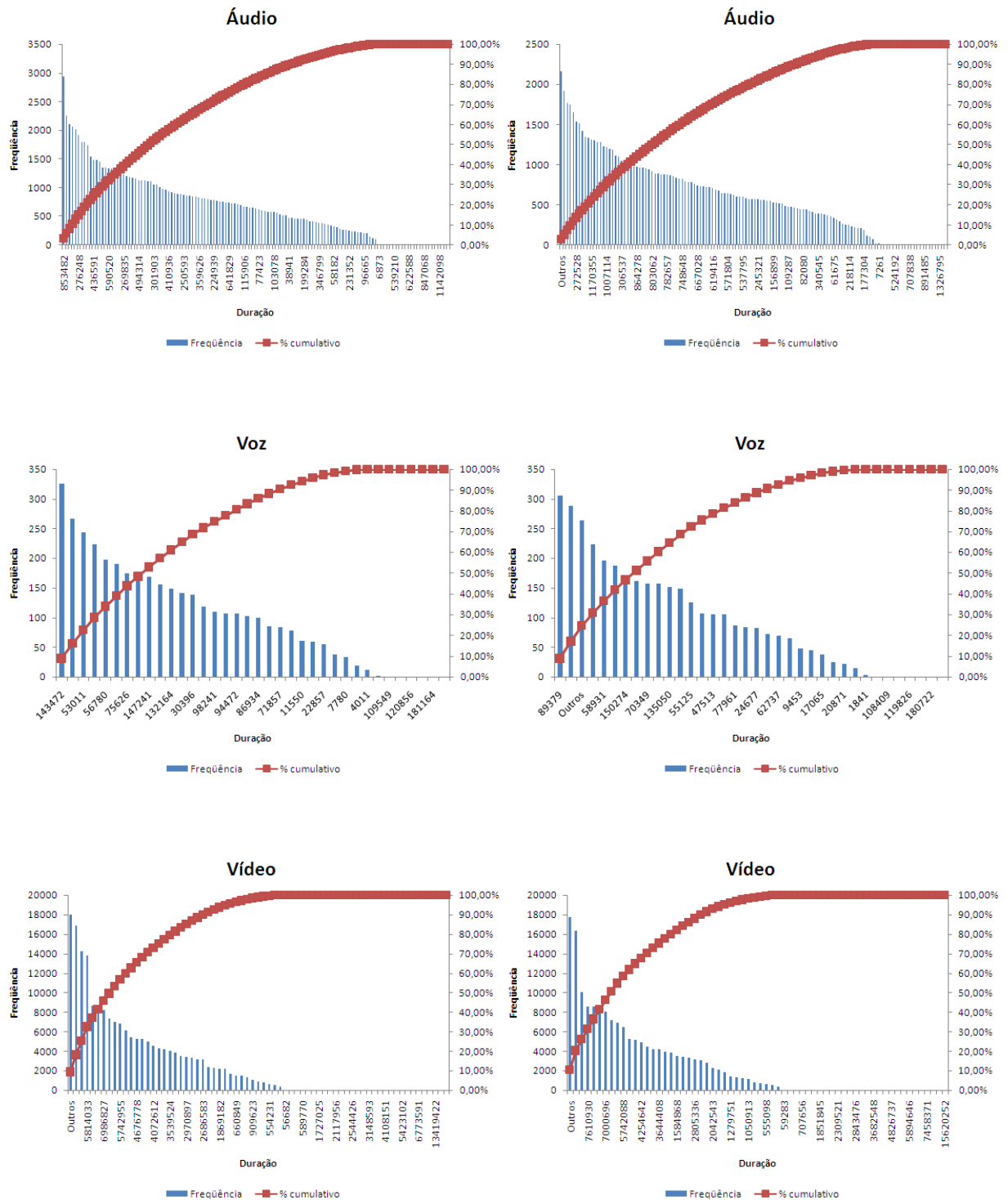
Diante desse cenário, quanto às avaliações experimentais propriamente ditas, as Figuras 29, 30, 31 e 32 ilustram os histogramas pertinentes aos fluxos de áudio, de voz e de vídeo, gerados por meio do gerador de tráfego multimídia e modelados por meio da função de distribuição exponencial, sem o intermédio do *gateway* multimídia (à esquerda) e com o intermédio do *gateway* multimídia (à direita), nesse caso, em transmissões de 30 minutos, 1 hora, 3 horas e 6 horas, respectivamente, no ambiente de emulação.

Por sua vez, quanto aos fluxos de tráfego gerados por meio do iPerf, os resultados obtidos em termos de tempos de transmissão são apresentados na Tabela 5; os resultados obtidos em termos de variações do atraso (*jitter*) são apresentados na Tabela 6; e, por sua vez, os resultados obtidos em termos de percentuais de datagramas perdidos são apresentados na Tabela 7. Em ambos os casos, os resultados obtidos são pertinentes às 5 transmissões para os fluxos de tráfego aleatórios descritos anteriormente, nesse caso, sem o intermédio do *gateway* multimídia, bem como com o intermédio do mesmo no ambiente de emulação.

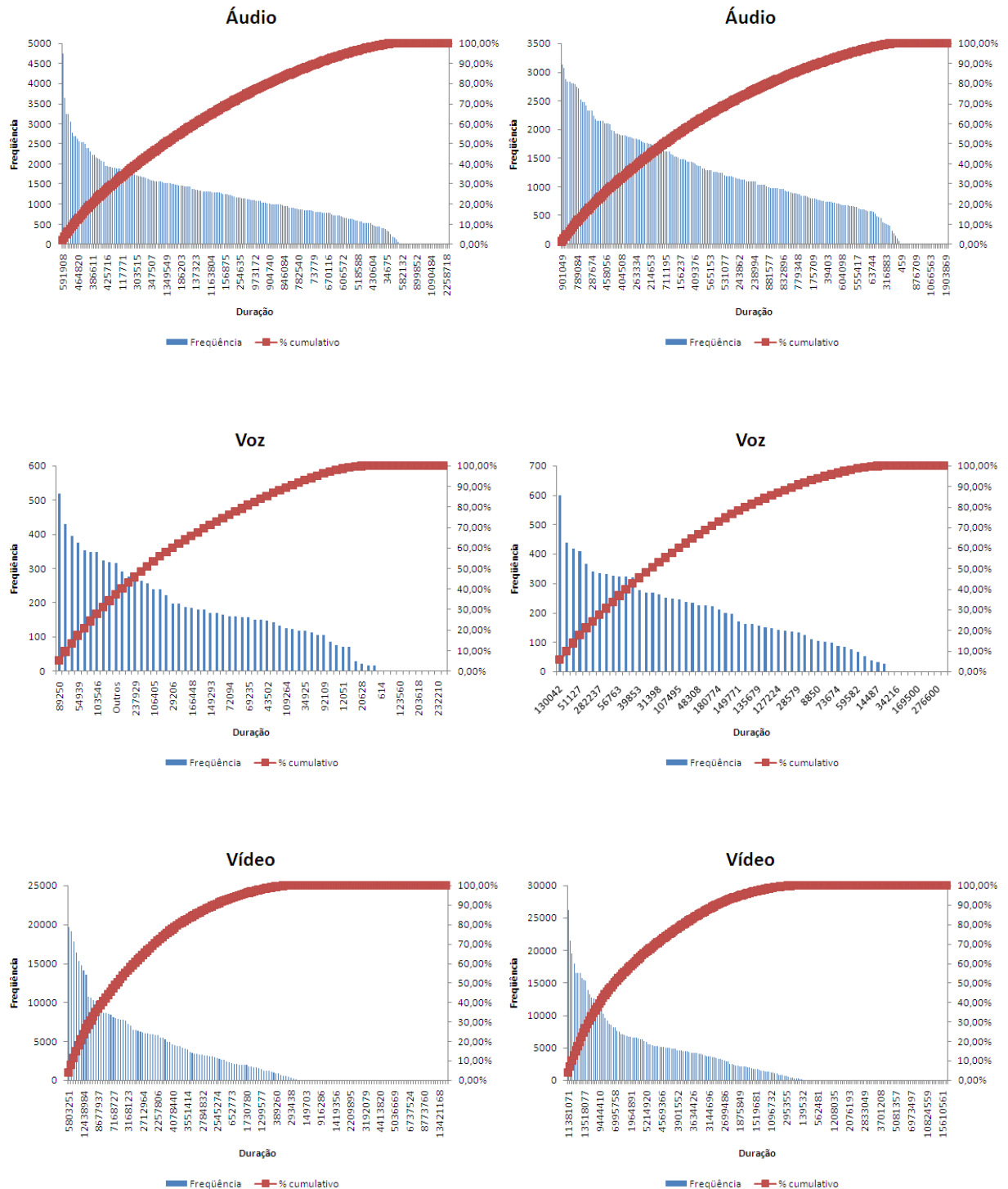
**Figura 29.** Histogramas pertinentes aos fluxos de áudio, de voz e de vídeo, gerados por meio do gerador de tráfego multimídia e modelados por meio da função de distribuição exponencial, sem o intermédio do gateway multimídia (à esquerda) e com o intermédio do gateway multimídia (à direita), em transmissões de 30 minutos.



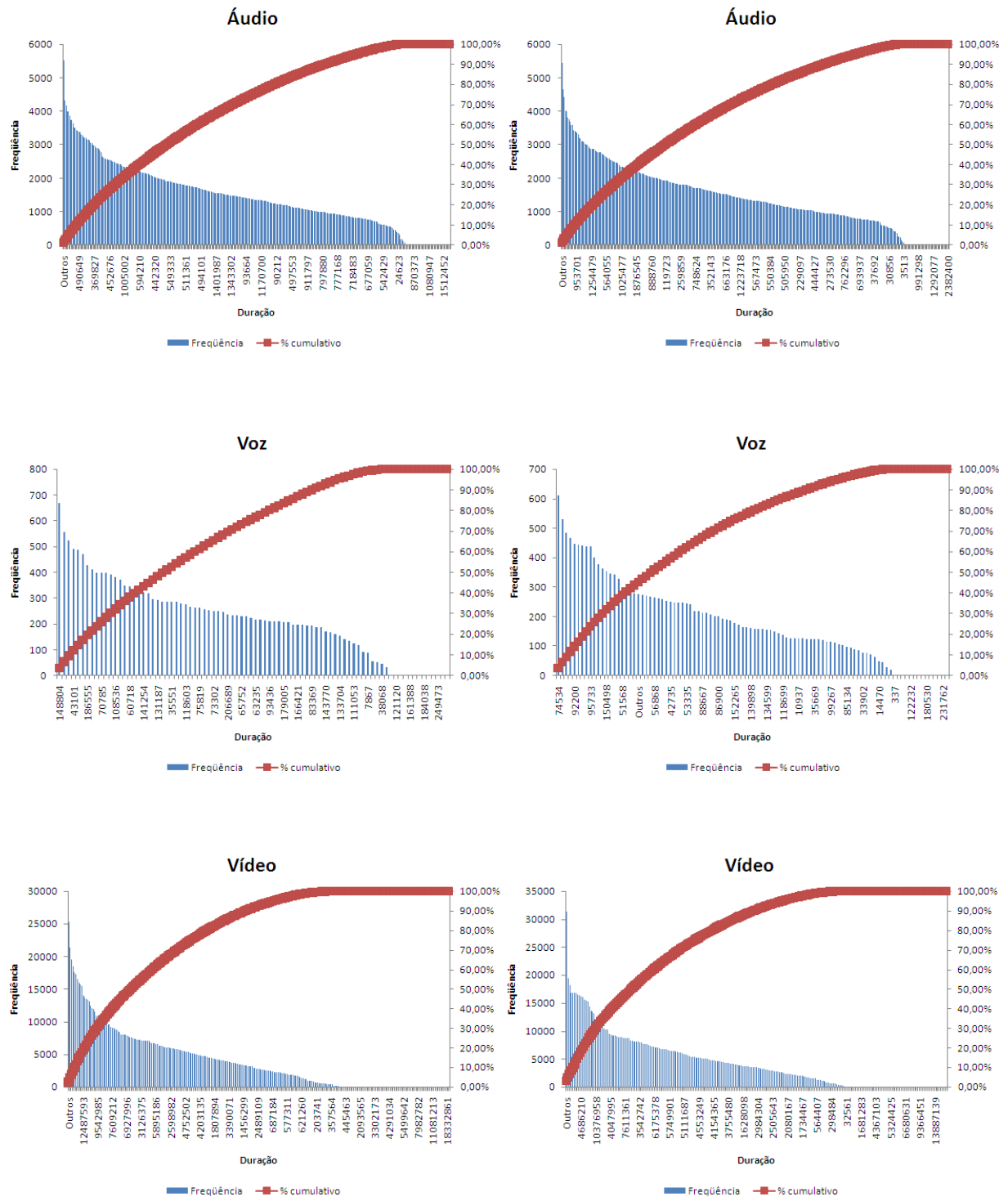
**Figura 30.** Histogramas pertinentes aos fluxos de áudio, de voz e de vídeo, gerados por meio do gerador de tráfego multimídia e modelados por meio da função de distribuição exponencial, sem o intermédio do gateway multimídia (à esquerda) e com o intermédio do gateway multimídia (à direita), em transmissões de 1 hora.



**Figura 31.** Histogramas pertinentes aos fluxos de áudio, de voz e de vídeo, gerados por meio do gerador de tráfego multimídia e modelados por meio da função de distribuição exponencial, sem o intermédio do gateway multimídia (à esquerda) e com o intermédio do gateway multimídia (à direita), em transmissões de 3 horas.



**Figura 32.** Histogramas pertinentes aos fluxos de áudio, de voz e de vídeo, gerados por meio do gerador de tráfego multimídia e modelados por meio da função de distribuição exponencial, sem o intermédio do gateway multimídia (à esquerda) e com o intermédio do gateway multimídia (à direita), em transmissões de 6 horas.



**Tabela 5.** Tempos de transmissão para os fluxos de tráfego gerados por meio do iPerf, sobre o protocolo UDP, sem o intermédio do *gateway* multimídia e com o intermédio do *gateway* multimídia.

Tráfego UDP de	Tempos de transmissão (em segundos)											
	Sem o intermédio do <i>gateway</i> multimídia						Com o intermédio do <i>gateway</i> multimídia					
	T1	T2	T3	T4	T5	Média	T1	T2	T3	T4	T5	Média
<b>225 MB</b>	1800	1800	1800	1800	1800	<b>1800</b>	1800	1800	1800	1800	1800	<b>1800</b>
<b>450 MB</b>	3600	3600	3600	3600	3600	<b>3600</b>	3600	3600	3600	3600	3600	<b>3600</b>
<b>1350 MB</b>	10800	10800	10800	10800	10800	<b>10800</b>	10800	10800	10800	10800	10800	<b>10800</b>
<b>2700 MB</b>	21600	21600	21600	21600	21600	<b>21600</b>	21600	21600	21600	21600	21600	<b>21600</b>

**Tabela 6.** Variações do atraso (*jitter*) para os fluxos de tráfego gerados por meio do iPerf, sobre o protocolo UDP, sem o intermédio do *gateway* multimídia e com o intermédio do *gateway* multimídia.

Tráfego UDP de	<i>Jitter</i> (em milissegundos)											
	Sem o intermédio do <i>gateway</i> multimídia						Com o intermédio do <i>gateway</i> multimídia					
	T1	T2	T3	T4	T5	Média	T1	T2	T3	T4	T5	Média
<b>225 MB</b>	6,29	6,45	7,48	5,87	7,92	<b>6,80</b>	5,68	3,59	4,84	6,21	5,07	<b>5,08</b>
<b>450 MB</b>	8,98	8,29	6,55	5,37	5,97	<b>7,03</b>	8,84	7,55	6,15	8,06	8,00	<b>7,72</b>
<b>1350 MB</b>	7,55	6,09	10,79	8,31	8,53	<b>8,25</b>	7,11	10,37	8,31	6,80	8,60	<b>8,24</b>
<b>2700 MB</b>	7,82	7,61	5,73	7,39	10,85	<b>7,88</b>	6,41	7,57	6,67	10,34	8,79	<b>7,96</b>

**Tabela 7.** Percentuais de datagramas perdidos para os fluxos de tráfego gerados por meio do iPerf, sobre o protocolo UDP, sem o intermédio do *gateway* multimídia e com o intermédio do *gateway* multimídia.

Tráfego UDP de	Percentual de datagramas perdidos											
	Sem o intermédio do <i>gateway</i> multimídia						Com o intermédio do <i>gateway</i> multimídia					
	T1	T2	T3	T4	T5	Média	T1	T2	T3	T4	T5	Média
<b>225 MB</b>	0,01	0,01	0,01	0,01	0,01	<b>0,01</b>	0,01	0,01	0,00	0,01	0,01	<b>0,01</b>
<b>450 MB</b>	0,01	0,01	0,01	0,01	0,01	<b>0,01</b>	0,01	0,01	0,02	0,02	0,01	<b>0,01</b>
<b>1350 MB</b>	0,01	0,01	0,01	0,01	0,01	<b>0,01</b>	0,01	0,01	0,01	0,01	0,01	<b>0,01</b>
<b>2700 MB</b>	0,01	0,02	0,01	0,02	0,01	<b>0,01</b>	0,02	0,01	0,01	0,02	0,01	<b>0,01</b>

Nesse contexto, quanto aos histogramas ilustrados anteriormente, é possível observar que, em ambas as experimentações, os resultados obtidos foram muito similares entre si. Em linhas gerais, seja sem o intermédio do *gateway* multimídia ou com o intermédio do mesmo, nas transmissões observadas de 30 minutos, 1 hora, 3 horas e 6 horas, os fluxos de



áudio, de voz e de vídeo, gerados por meio do gerador de tráfego multimídia e modelados por meio da função de distribuição exponencial, mantiveram suas respectivas características de tráfego (no âmbito de sua modelagem) no ambiente de emulação. Essas observações são especialmente importantes para identificar eventuais alterações quanto a tais características durante o processo de encaminhamento diferenciado fornecido pelo *gateway* multimídia, as quais não foram identificadas durante tais experimentações.

Por sua vez, quanto aos fluxos de tráfego gerados por meio do iPerf, assim como quanto aos histogramas pertinentes aos fluxos gerados por meio do gerador de tráfego multimídia, os resultados obtidos em ambas as experimentações também foram muito similares entre si. Como exemplo, é possível observar que os tempos de transmissão sem o intermédio do *gateway* multimídia, bem como com o intermédio do mesmo, foram exatamente os mesmos em todos os testes de ambas as experimentações. Nesse caso, os tempos de transmissão para os fluxos de 225 MB foram de 1800 segundos em ambos os casos, os tempos de transmissão para os fluxos de 450 MB foram de 3600 segundos em ambos os casos, os tempos de transmissão para os fluxos de 1350 MB foram de 10800 segundos em ambos os casos e, por sua vez, os tempos de transmissão para os fluxos de 2700 MB foram de 21600 segundos em ambos os casos. Quanto ao *jitter* observado em tais transmissões, os valores médios obtidos também foram muito similares em ambos os casos, com distinções inferiores a 1 milissegundo nos resultados médios das transmissões de 450 MB, 1350 MB e 2700 MB e de menos de 2 milissegundos nas transmissões de 225 MB. Em tais experimentações, é possível observar uma pequena diferença média de tempo (à menor) nos fluxos não intermediados pelo *gateway* multimídia nas transmissões de 450 MB e de 2700 MB (diferença média de 0,69 milissegundos para os fluxos de 450 MB e de 0,08 milissegundos para os fluxos de 2700 MB). Por sua vez, nas transmissões de 225 MB e de 1350 MB, é possível observar uma pequena diferença de tempo (à menor) nos fluxos intermediados pelo *gateway* multimídia (diferença média de 1,72 milissegundos para os fluxos de 225 MB e de 0,01 milissegundos para os fluxos de 1350 MB). De modo complementar, é possível observar que, na maioria das experimentações individuais, o *jitter* obtido apresentou resultados numa faixa entre 6 e 9 milissegundos (em 14 dos 20 testes sem o intermédio do *gateway* multimídia e, também, em 14 dos 20 testes com o intermédio do *gateway* multimídia), em que o menor valor obtido foi de 3,59 milissegundos (transmissão T2 de 225 MB com o intermédio do *gateway* multimídia) e o maior valor obtido foi de 10,85 milissegundos (transmissão T5 de 2700 MB sem o intermédio do *gateway* multimídia). Por

sua vez, quanto ao percentual de datagramas perdidos em tais transmissões, os resultados obtidos também foram muito similares em ambos os casos, com resultados médios de 0,01% em todas as experimentações. Nesse caso, as principais distinções são pertinentes a 2 testes individuais sem o intermédio do *gateway* multimídia (testes T2 e T4 para os fluxos de 2700 MB) e 4 testes individuais com o *gateway* multimídia (testes T3 e T4 para os fluxos de 450 MB e testes T1 e T4 para os fluxos de 2700 MB), os quais apresentaram perda de 0,02% em suas transmissões. Nos demais testes individuais, a perda foi de 0,01% em todas as transmissões.

Nesse contexto, ainda quanto a perda de pacotes observada quanto aos fluxos de tráfego gerados por meio do iPerf, apenas para efeito de registro, observou-se que, em linhas gerais, tais perdas ocorreram em momentos específicos das transmissões, nesse caso, comumente em intervalos de cerca de 600 segundos. Essas observações foram identificadas nas experimentações sem o emprego (intermédio) do *gateway* multimídia, bem como com o emprego do mesmo no ambiente de emulação. Diante desse cenário, para efeitos de curiosidade e pesquisa, observou-se a ocorrência de requisições ARP (*Address Resolution Protocol*) partindo dos *hosts* transmissores do iPerf em tais intervalos de tempo. Em linhas gerais, essas requisições buscavam identificar o endereço MAC associado ao endereço IP do *gateway* de acesso à rede do respectivo *host* transmissor do iPerf. Essa mesma observação também foi identificada em algumas perdas no início das transmissões geradas pelo iPerf. Como exemplo a Figura 33 ilustra dois momentos em que houve a ocorrência de perdas de pacotes junto aos fluxos de tráfego gerados por meio do iPerf entre o cliente multimídia H3 e o servidor multimídia H1.

**Figura 33.** Trecho de *log* junto ao servidor multimídia H1 ilustrando a ocorrência de perdas de pacotes nos tempos de transmissão de 608-609 segundos e de 1210-1211 segundos.

```
[...]
[ 6] 608.00-609.00   sec 128 KBytes   1.05 Mbits/sec 4.644 ms 1/17 (5.9%)
[...]
[ 6] 1210.00-1211.00 sec 96.0 KBytes   786 Kbits/sec 6.618 ms 2/14 (14%)
[...]
```

Nesse exemplo, as perdas observadas foram de 1 pacote no tempo de transmissão de 608-609 segundos e de 2 pacotes no tempo de transmissão de 1210-1211 segundos. Em tais momentos, foi possível identificar requisições ARP originadas pelo *host* H3 com o objetivo de identificar o endereço MAC de seu *gateway* na rede local (Figura 34).

**Figura 34.** Requisições ARP realizadas pelo *host* cliente H3 em dois momentos (A e B) com perdas de pacotes observadas no ambiente de emulação.

(A)

No.	Time	Source	Destination	Protocol	Length	Info
2049	611.497966000	CadmusCo_ad:53:a6	Broadcast	OF 1.3	168	of_packet_in
2050	611.498052000	192.168.56.1	192.168.56.200	TCP	66	openflow > 52299 [ACK] Seq=8327 Ack=8301 Win=459 Len=0 TSval=1294530 TSecr=
2051	611.498152000	CadmusCo_ad:53:a6	Broadcast	OF 1.3	166	of_packet_out
2052	611.498432000	CadmusCo_af:1d:63	CadmusCo_ad:53:a6	OF 1.3	150	of_packet_in
2053	611.499220000	192.168.56.1	192.168.56.200	OF 1.3	178	of_flow_add
2054	611.499249000	CadmusCo_af:1d:63	CadmusCo_ad:53:a6	OF 1.3	148	of_packet_out
2055	611.499471000	192.168.56.200	192.168.56.1	TCP	66	52299 > openflow [ACK] Seq=8385 Ack=8621 Win=354 Len=0 TSval=1157730 TSecr=

(B)

No.	Time	Source	Destination	Protocol	Length	Info
4045	1214.416085000	CadmusCo_ad:53:a6	Broadcast	OF 1.3	168	of_packet_in
4046	1214.416407000	192.168.56.1	192.168.56.200	TCP	66	openflow > 52299 [ACK] Seq=15875 Ack=15825 Win=475 Len=0 TSval=1445262 TSecr=
4047	1214.416555000	CadmusCo_ad:53:a6	Broadcast	OF 1.3	166	of_packet_out
4048	1214.417447000	CadmusCo_af:1d:63	CadmusCo_ad:53:a6	OF 1.3	150	of_packet_in
4049	1214.420248000	192.168.56.1	192.168.56.200	OF 1.3	178	of_flow_add
4050	1214.420318000	CadmusCo_af:1d:63	CadmusCo_ad:53:a6	OF 1.3	148	of_packet_out
4051	1214.421095000	192.168.56.200	192.168.56.1	TCP	66	52299 > openflow [ACK] Seq=15909 Ack=16169 Win=363 Len=0 TSval=1308460 TSecr=

Nesses exemplos, considerando o tempo de inicialização do analisador de protocolos Wireshark, as requisições ARP foram identificadas nos tempos iniciais de 611 segundos (item A da Figura 34) e 1214 segundos (item B da Figura 34), nos “mesmos momentos” (intervalos de cerca de 600 segundos) em que ocorreram as perdas de pacotes ilustradas na Figura 33. Ainda assim, é importante destacar que nenhum estudo específico foi

realizado com o objetivo de mapear ou correlacionar as perdas de pacotes com algum evento ou situação específica da rede (tais como com essas requisições ARP, por exemplo), mas sim apenas para comparar tais perdas (além dos tempos de transmissão e *jitter*) em transmissões sem o emprego do *gateway* multimídia, bem como com o emprego do mesmo no ambiente de emulação. Por esse motivo, essas observações ficam apenas como registro das avaliações experimentais, as quais, se aplicáveis, podem ser melhores exploradas em um eventual contexto/objetivo específico para tal.

Diante desse cenário, com base nos resultados obtidos nas avaliações descritas nesta Seção, pode-se observar ampla similaridade entre os fluxos de tráfego não intermediados pelo *gateway* multimídia, bem como intermediados por ele no ambiente de emulação (nesse caso, em termos de características quanto à modelagem de tráfego, tempos de transmissão, variações do atraso e percentual de datagramas perdidos). Em linhas gerais, essas observações são especialmente importantes para identificar eventuais alterações durante o processo de encaminhamento diferenciado fornecido pelo *gateway* multimídia, as quais não foram identificadas durante tais experimentações.

#### **4.2.4 – Encaminhamento por Fluxos com Suporte à QoS**

Para avaliar o recurso de QoS durante o processo de encaminhamento por fluxos junto ao *gateway* multimídia, três serviços multimídia comumente explorados no âmbito Internet foram empregados no ambiente de emulação: um serviço de *streaming* de áudio e vídeo (embasado nos protocolos RTP e RTSP), um serviço de transferência de arquivos (embasado no protocolo FTP) e um serviço de hospedagem *web* (embasado no protocolo HTTP). Por meio de tais serviços, configurados junto ao servidor multimídia H1, três diferentes tipos de conteúdos multimídia foram disponibilizados aos clientes da rede: um conteúdo multimídia de áudio (arquivo de referência: *sample.mp3*, formato MP3, duração de 4m06s e tamanho de 10.040.544 *bytes*), um conteúdo multimídia de vídeo (arquivo de referência: *sample.wmv*, formato WMV, resolução 1280x532, duração de 1m48s e tamanho de 46.897.701 *bytes*) e um conteúdo multimídia de dados (arquivo de referência: *sample.zip*, formato ZIP e tamanho de 155.104.886 *bytes*). Por sua vez, o acesso a tais conteúdos foi realizado por meio dos clientes multimídia H3 (conteúdo multimídia de vídeo – *streaming* de

vídeo sobre RTP e RTSP), H4 (conteúdo multimídia de dados – arquivo de dados sobre FTP e sobre HTTP) e H5 (conteúdo multimídia de áudio – *streaming* de áudio sobre RTP e RTSP).

Nesse cenário, os fluxos de tráfego pertinentes a transmissão de cada um desses conteúdos multimídia foram devidamente rotulados, por meio do adaptador de tráfego multimídia, com seus respectivos identificadores de serviços multimídia, nesse caso, identificadores DSCP 0x03 (conteúdo de dados sobre HTTP), 0x04 (conteúdo de dados sobre FTP), DSCP 0x13 (conteúdo de áudio sobre RTP e RTSP) e 0x33 (conteúdo de vídeo sobre RTP e RTSP). Dessa forma, por meio de tais identificadores, o *gateway* multimídia é capaz de distinguir e fornecer diferentes níveis de QoS a tais fluxos de tráfego multimídia, possibilitando, por exemplo, melhores níveis de QoS aos fluxos de *streaming* de áudio e/ou vídeo ou aos fluxos de dados dos serviços de transferência de arquivos e/ou de hospedagem *web*, de modo geral.

Diante desse cenário, quatro experimentações foram realizadas no ambiente de emulação:

- (I) Transmissões multimídia de áudio, de vídeo e de dados, sem o emprego de recursos de QoS junto ao *gateway* multimídia, e com ambos os conteúdos sendo acessados individualmente na rede;
- (II) Transmissões multimídia de áudio, de vídeo e de dados, sem o emprego de recursos de QoS junto ao *gateway* multimídia, e com ambos os conteúdos sendo acessados simultaneamente na rede;
- (III) Transmissões multimídia de áudio, de vídeo e de dados, com o emprego de recursos de QoS junto ao *gateway* multimídia, e com ambos os conteúdos sendo acessados individualmente na rede;
- (IV) Transmissões multimídia de áudio, de vídeo e de dados, com o emprego de recursos de QoS junto ao *gateway* multimídia, e com ambos os conteúdos sendo acessados simultaneamente na rede.

Nesse cenário, para cada experimentação, 5 transmissões de áudio, de vídeo, de dados sobre FTP e de dados sobre HTTP foram consideradas (transmissões de T1 à T5), ambas realizadas por seu respectivo cliente multimídia e, em ambos os casos, utilizando os mesmos arquivos de referência. Em tal cenário, em um primeiro momento, nenhuma configuração no âmbito da QoS foi empregada junto ao *gateway* multimídia na rede local. Por

sua vez, cada conteúdo multimídia em questão foi acessado individualmente na rede. Em seguida, ainda sem configurações no âmbito da QoS, ambos os conteúdos multimídia foram acessados simultaneamente, nesse caso, com cada conteúdo em específico sendo requisitado por seu respectivo cliente multimídia. Ao final, empregando as definições de QoS junto ao *gateway* multimídia, repetiu-se a mesma estratégia adotada anteriormente, com ambos os conteúdos multimídia sendo acessados individualmente em um primeiro momento e simultaneamente em um segundo momento.

Nesse contexto, a Tabela 8 apresenta as taxas de transmissão para os fluxos pertinentes aos conteúdos multimídia de áudio, de vídeo e de dados, sem o emprego de recursos de QoS junto ao *gateway* multimídia, em transmissões individuais e em transmissões simultâneas de ambos no ambiente de emulação.

**Tabela 8.** Taxas de transmissão para os fluxos pertinentes aos conteúdos multimídia de áudio, de vídeo e de dados, sem o emprego de recursos de QoS junto ao *gateway* multimídia, em transmissões individuais e em transmissões simultâneas de ambos no ambiente de emulação.

Taxas de transmissão para os conteúdos multimídia de áudio, de vídeo e de dados (sem o emprego de recursos de QoS junto ao <i>gateway</i> multimídia)												
Fluxos de	Transmissões individuais						Transmissões simultâneas					
	T1	T2	T3	T4	T5	Média	T1	T2	T3	T4	T5	Média
<b>Áudio</b>	1,68 Mbps	1,68 Mbps	1,67 Mbps	1,68 Mbps	1,67 Mbps	<b>1,68</b> <b>Mbps</b>	1,64 Mbps	1,65 Mbps	1,65 Mbps	1,64 Mbps	1,65 Mbps	<b>1,65</b> <b>Mbps</b>
<b>Vídeo</b>	3,29 Mbps	3,32 Mbps	3,28 Mbps	3,30 Mbps	3,34 Mbps	<b>3,31</b> <b>Mbps</b>	2,04 Mbps	2,05 Mbps	2,06 Mbps	2,03 Mbps	2,06 Mbps	<b>2,05</b> <b>Mbps</b>
<b>Dados (FTP)</b>	9,48 Mbps	9,50 Mbps	9,52 Mbps	9,54 Mbps	9,51 Mbps	<b>9,51</b> <b>Mbps</b>	4,20 Mbps	4,21 Mbps	4,21 Mbps	4,22 Mbps	4,22 Mbps	<b>4,21</b> <b>Mbps</b>
<b>Dados (HTTP)</b>	8,80 Mbps	8,80 Mbps	8,88 Mbps	8,80 Mbps	8,88 Mbps	<b>8,83</b> <b>Mbps</b>	3,99 Mbps	3,99 Mbps	4,00 Mbps	4,03 Mbps	4,02 Mbps	<b>4,01</b> <b>Mbps</b>

Diante dos resultados obtidos, é possível observar que, individualmente e sem a aplicação de recursos de QoS junto ao *gateway* multimídia, as taxas médias de transmissão observadas para as 5 transmissões pertinentes aos conteúdos multimídia de áudio (*streaming* de áudio), de vídeo (*streaming* de vídeo), de dados sobre FTP e de dados sobre HTTP foram de, respectivamente, 1,68 Mbps, 3,31 Mbps, 9,51 Mbps e 8,83 Mbps. Por outro lado, no acesso simultâneo aos mesmos conteúdos multimídia na rede, suas taxas de transmissão foram inferiores. Nesse caso, as taxas médias de transmissão observadas para ambos os conteúdos multimídia foram de 1,65 Mbps (*streaming* de áudio), 2,05 Mbps (*streaming* de

vídeo), 4,21 Mbps (dados sobre FTP) e 4,01 Mbps (dados sobre HTTP). Nesse contexto, é importante ressaltar que, no ambiente de emulação, o acesso a ambos os conteúdos multimídia na “Internet” (nesse caso, acessos entre os clientes multimídia e os servidores multimídia) é realizado por meio de um *link* de transmissão na ordem de 10 Mbps, fornecido pelo roteador R1, tal como ilustrado anteriormente na Figura 15.

Nesse cenário, essa redução nas taxas de transmissão de ambos os conteúdos pode afetar negativamente a percepção de QoE do usuário final, em especial, quanto às aplicações pertinentes aos *streamings* de áudio e de vídeo, tais como por meio de eventuais *delays* pertinentes à *bufferizações* junto aos *players* dos clientes multimídia ou por meio de eventuais degradações de qualidade de áudio e/ou vídeo pertinentes à perdas de pacotes, dentre outros.

Diante desse cenário, objetivando o fornecimento de melhores níveis de QoS e, também, de QoE, em especial, aos fluxos de *streaming* de áudio e de vídeo em transmissões simultâneas aos fluxos de dados na rede, cinco classes de tráfego, com diferentes configurações em termos de prioridade e alocações de banda foram configuradas junto ao *gateway* multimídia: uma classe com prioridade 1 para os fluxos de vídeo (*streaming* de vídeo), uma classe com prioridade 2 para os fluxos de áudio (*streaming* de áudio), uma classe com prioridade 3 para os fluxos de dados sobre HTTP, uma classe com prioridade 4 para os fluxos de dados sobre FTP e uma classe com prioridade 5 para os demais fluxos de tráfego da rede. Dessas, a classe com prioridade 1 é a que possui maior prioridade e a classe com prioridade 5 é a que possui menor prioridade. Essa priorização entre uma e outra classe de tráfego e seus respectivos fluxos de tráfego é realizada por meio dos mecanismos de escalonamento empregados junto ao *gateway* multimídia, nesse caso, fornecidos pelo HTB durante tal experimentação.

Nesse contexto, cada classe de tráfego recebeu diferentes configurações em termos de alocações de banda, em uma proporção de até 60% para os fluxos de *streaming* de áudio e de vídeo e de até 40% para os demais fluxos de tráfego de rede (fluxos de dados e eventuais outros fluxos de tráfego). Para tal, as classes pertinentes aos fluxos áudio e de vídeo receberam alocações de 2 Mbps e de 4 Mbps, respectivamente, podendo compartilhar recursos de banda entre si até 6 Mbps (de acordo com a prioridade de cada classe de tráfego em questão). Por sua vez, as classes pertinentes aos fluxos de dados sobre FTP, dados sobre HTTP e eventuais outros fluxos de tráfego de rede receberam alocações de 1 Mbps cada, podendo compartilhar recursos de banda entre si até 4 Mbps (de acordo com a prioridade de cada classe de tráfego em questão).

Nesse cenário, a associação entre os fluxos de tráfego e suas respectivas classes de tráfego foi realizada com base nos identificadores DSCP pertinentes a cada fluxo de tráfego em questão. Nesse caso, fluxos com identificadores DSCP distintos aos definidos aos fluxos de áudio, de vídeo, de dados sobre FTP e de dados sobre HTTP foram automaticamente tidos como “outros fluxos de tráfego de rede”.

Nesse contexto, é importante destacar que os critérios e estratégias para a definição e configuração das classes de tráfego, bem como das prioridades e alocações quanto aos seus respectivos fluxos de tráfego, podem ser configurados de acordo com cada ambiente e necessidade em questão. Dessa forma, os critérios e estratégias empregados nesta Seção possuem apenas caráter avaliativo experimental, passíveis de eventuais ajustes e adequações se assim forem necessários e aplicáveis.

Diante desse cenário, a Tabela 9 apresenta as taxas de transmissão para os fluxos pertinentes aos conteúdos multimídia de áudio, de vídeo e de dados, com o emprego de recursos de QoS junto ao *gateway* multimídia, em transmissões individuais e em transmissões simultâneas de ambos no ambiente de emulação.

**Tabela 9.** Taxas de transmissão para os fluxos pertinentes aos conteúdos multimídia de áudio, de vídeo e de dados, com o emprego de recursos de QoS junto ao *gateway* multimídia, em transmissões individuais e em transmissões simultâneas de ambos no ambiente de emulação.

Taxas de transmissão para os conteúdos multimídia de áudio, de vídeo e de dados (com o emprego de recursos de QoS junto ao <i>gateway</i> multimídia)												
Fluxos de	Transmissões individuais						Transmissões simultâneas					
	T1	T2	T3	T4	T5	Média	T1	T2	T3	T4	T5	Média
<b>Áudio</b>	1,67 Mbps	1,67 Mbps	1,70 Mbps	1,71 Mbps	1,68 Mbps	<b>1,69</b> <b>Mbps</b>	1,68 Mbps	1,71 Mbps	1,68 Mbps	1,67 Mbps	1,68 Mbps	<b>1,68</b> <b>Mbps</b>
<b>Vídeo</b>	3,30 Mbps	3,26 Mbps	3,29 Mbps	3,31 Mbps	3,26 Mbps	<b>3,28</b> <b>Mbps</b>	3,29 Mbps	3,26 Mbps	3,21 Mbps	3,27 Mbps	3,28 Mbps	<b>3,26</b> <b>Mbps</b>
<b>Dados (FTP)</b>	3,84 Mbps	3,85 Mbps	3,85 Mbps	3,84 Mbps	3,85 Mbps	<b>3,85</b> <b>Mbps</b>	1,92 Mbps	1,92 Mbps	1,90 Mbps	1,92 Mbps	1,92 Mbps	<b>1,92</b> <b>Mbps</b>
<b>Dados (HTTP)</b>	3,74 Mbps	3,74 Mbps	3,74 Mbps	3,56 Mbps	3,55 Mbps	<b>3,67</b> <b>Mbps</b>	2,81 Mbps	2,81 Mbps	2,66 Mbps	2,79 Mbps	2,79 Mbps	<b>2,77</b> <b>Mbps</b>

Diante dos resultados obtidos, é possível observar que, em ambos os casos, as taxas de transmissão pertinentes aos conteúdos multimídia em questão se mantiveram de acordo com as definições de suas respectivas classes de tráfego junto ao *gateway* multimídia. Nesse caso, é possível observar que as taxas médias de transmissão observadas para as 5 transmissões individuais pertinentes aos conteúdos multimídia de áudio (*streaming* de áudio),

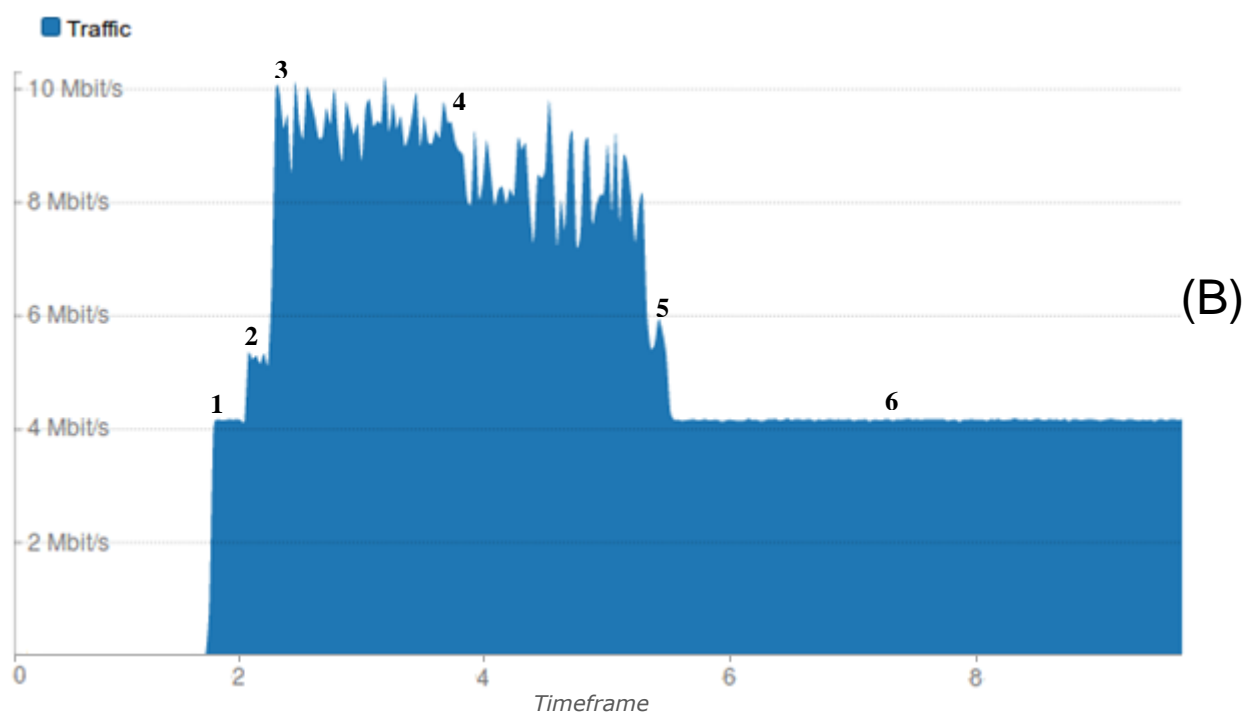
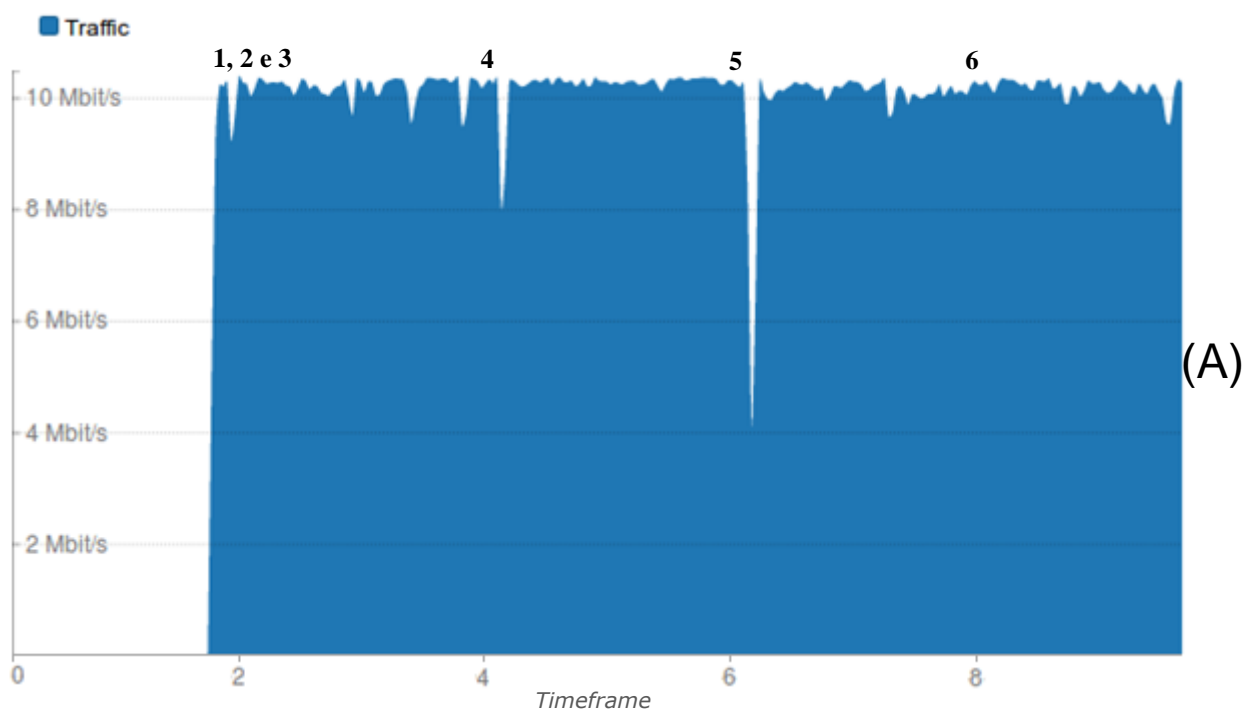


de vídeo (*streaming* de vídeo), de dados sobre FTP e de dados sobre HTTP foram de, respectivamente, 1,69 Mbps, 3,28 Mbps, 3,85 Mbps e 3,67 Mbps. Por sua vez, na transmissão simultânea de ambos os conteúdos multimídia na rede, as taxas médias de transmissão observadas foram de 1,68 Mbps para os fluxos do *streaming* de áudio, 3,26 Mbps para os fluxos do *streaming* de vídeo, 1,92 Mbps para os fluxos de dados sobre FTP e 2,77 Mbps para os fluxos de dados sobre HTTP.

Nesse contexto, observe que, em especial aos fluxos pertinentes aos *streamings* de áudio e de vídeo, suas taxas de transmissão foram muito similares tanto nas transmissões individuais, quanto nas transmissões simultâneas de ambos em conjunto com os fluxos de dados, nesse caso, na ordem aproximada de 1,70 Mbps e de 3,30 Mbps, respectivamente. Por sua vez, quanto aos fluxos de dados, observe que, nas transmissões individuais, ambos estenderam seus escopos de 1 Mbps para até os 4 Mbps possíveis para tais, nesse caso, com taxas na ordem de 3,85 Mbps para os fluxos de dados sobre FTP e entre 3,55 Mbps e 3,74 Mbps para os fluxos de dados sobre HTTP. Da mesma forma, nas transmissões simultâneas de ambos na rede, observe que os fluxos de dados sobre HTTP foram devidamente priorizados no compartilhamento dos recursos de banda de rede sobre os fluxos de dados sobre FTP, com taxas na ordem de 2,80 Mbps e 1,90 Mbps, respectivamente.

Diante desse cenário, de modo complementar às Tabelas 8 e 9, no âmbito da utilização dos recursos de rede em tais experimentações, a Figura 35 ilustra duas transmissões simultâneas de ambos os fluxos de tráfego multimídia experimentais no ambiente de emulação, nesse caso, sem o emprego de recursos de QoS junto ao *gateway* multimídia (A) e com o emprego de recursos de QoS junto ao *gateway* multimídia (B). Em tais transmissões, alguns pontos (de 1 até 6) possibilitam a identificação dos momentos aos quais ambos os conteúdos multimídia foram acessados na rede. Nesse caso, o ponto 1 refere-se ao início da transmissão dos fluxos de dados (sobre FTP e sobre HTTP, respectivamente); o ponto 2 refere-se ao início das transmissões dos fluxos de vídeo; o ponto 3 refere-se ao início da transmissão dos fluxos de áudio; o ponto 4 refere-se ao término da transmissão dos fluxos de áudio; o ponto 5 refere-se ao término da transmissão dos fluxos de vídeo; e, por fim, o ponto 6 refere-se à continuidade das transmissões pertinentes aos fluxos de dados (sobre FTP e sobre HTTP).

**Figura 35.** Utilização dos recursos de rede em duas transmissões simultâneas pertinentes aos conteúdos multimídia de áudio, de vídeo e de dados no ambiente de emulação, nesse caso, sem o emprego de recursos de QoS junto ao *gateway* multimídia (A) e com emprego de recursos de QoS junto ao *gateway* multimídia (B).



Nesse cenário, no âmbito da QoE percebida pelo usuário final durante as transmissões simultâneas sem o emprego de recursos de QoS, os fluxos pertinentes ao *streaming* de vídeo foram os mais afetados. Em transmissões empregando o protocolo de transporte TCP, uma série de *delays* pertinentes a *bufferizações* foi observada durante sua exibição no *player* de vídeo do cliente multimídia. Por sua vez, em transmissões empregando o protocolo de transporte UDP, devido às maiores perdas de pacotes (acarretando, por exemplo, em maiores perdas de *frames*), observou-se uma maior degradação na qualidade do vídeo, ambas afetando negativamente a percepção de QoE do usuário final, de modo geral.

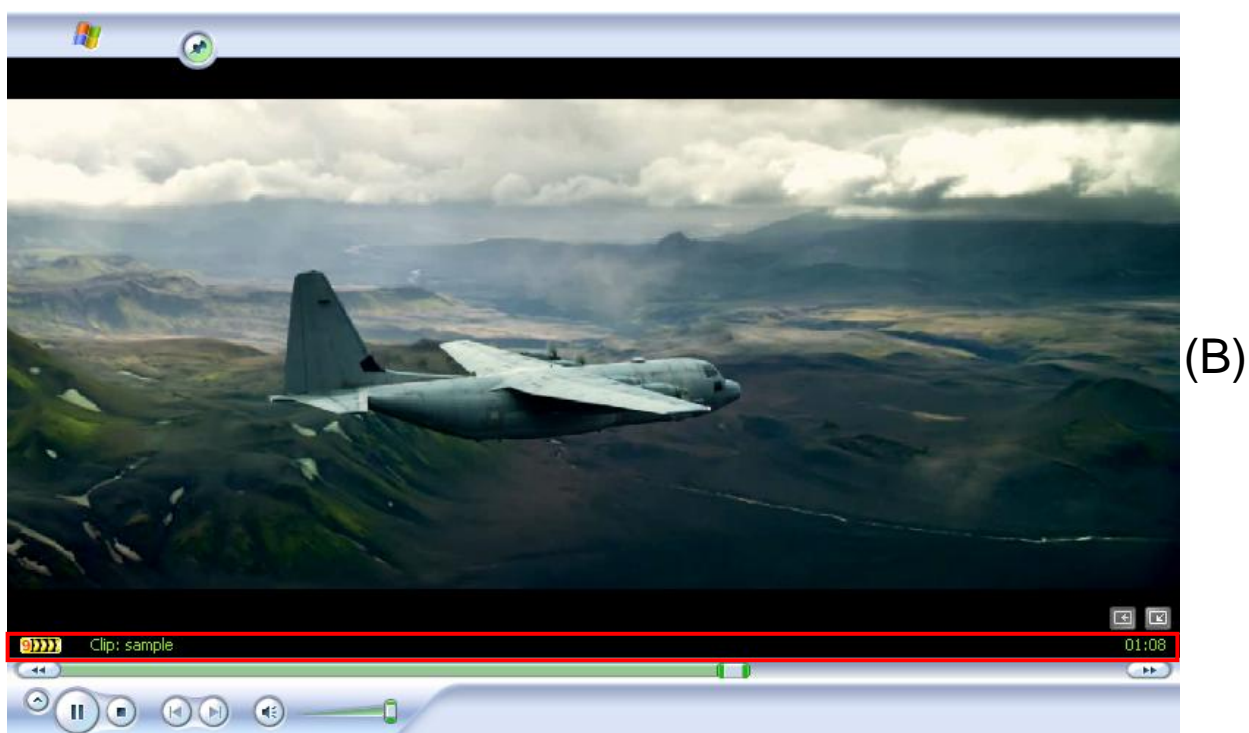
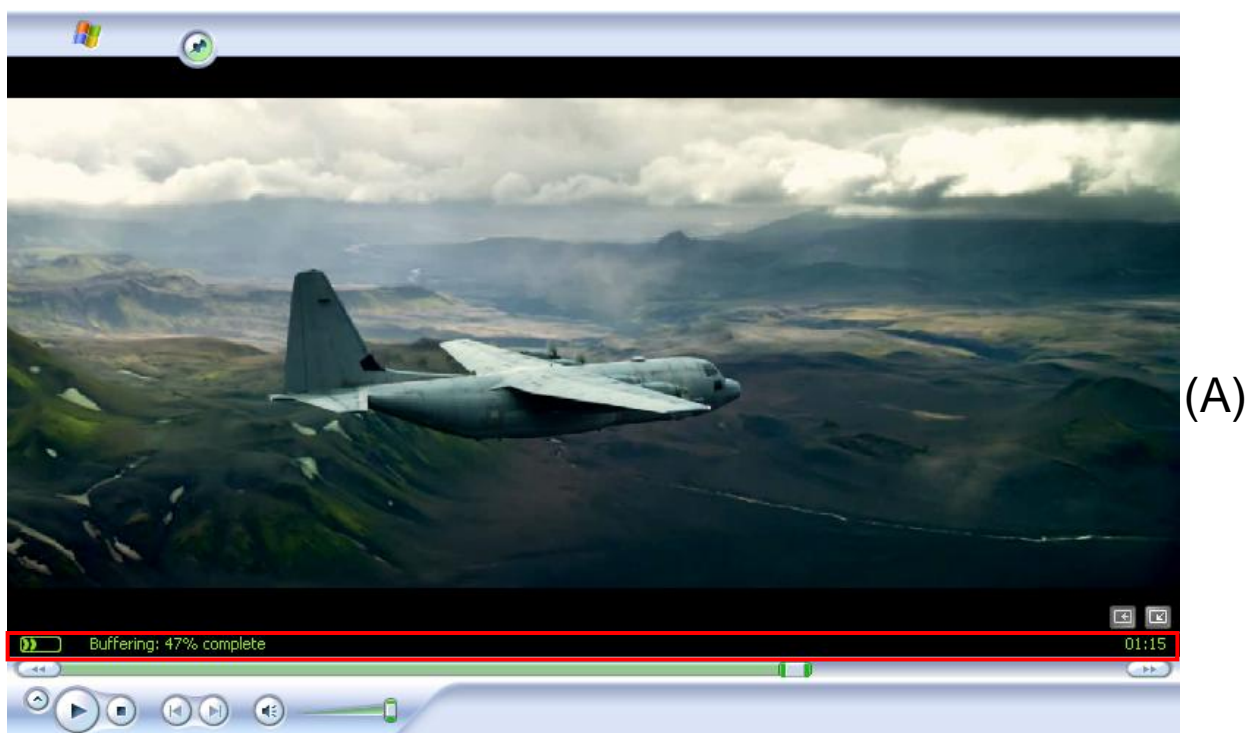
Por outro lado, o emprego de recursos de QoS junto ao *gateway* multimídia, por meio das distintas configurações em termos de classes de tráfego para cada conteúdo multimídia experimentado no ambiente de emulação, possibilitou contribuir para o aprimoramento da QoE percebida pelo usuário final, tanto por meio das observações descritas anteriormente, como por meio de métricas tradicionais de QoE, tal como por meio do PSNR (*Peak Signal-to-Noise Ratio*) (WINKLER; MOHANDAS, 2008), que estima a qualidade do vídeo em decibéis, comparando o vídeo original com o vídeo recebido pelo cliente final.

Nesse contexto, como exemplo, a Figura 36 ilustra dois momentos pertinentes à transmissão do *streaming* de vídeo no *player* do cliente multimídia, sem o emprego de recursos de QoS e com a ocorrência de *delays* pertinentes à *bufferizações* (A) e com o emprego de recursos de QoS e sem a ocorrência de tais *delays* (B). Por meio de tal ilustração, observe que o mesmo *frame* é exibido ao cliente final em tempos distintos, nesse caso, no momento 1m15s (com atraso) sem o emprego de recursos de QoS e no momento 1m08s com o emprego de recursos de QoS.

De modo complementar, quanto às observações pertinentes à qualidade do vídeo recebido pelo cliente final, a Figura 37 ilustra o *frame* de número 563 do vídeo de referência original (A) e o *frame* de mesmo número recebido pelo cliente final com o emprego de recursos de QoS (B) e sem o emprego de recursos de QoS (C). Por meio de tal ilustração, observe que o *frame* recebido com o emprego de recursos de QoS está de acordo com o *frame* original, por outro lado, o *frame* recebido sem o emprego de recursos de QoS é um *frame* degradado quando comparado ao *frame* original.

Por sua vez, enquanto métrica de QoE, a Figura 38 ilustra o PSNR pertinente ao vídeo recebido pelo cliente final sem o emprego de recursos de QoS (linha em vermelho) e com o emprego de recursos de QoS (linha em verde) junto ao *gateway* multimídia.

**Figura 36.** Exemplo de dois momentos pertinentes à transmissão do *streaming* de vídeo no *player* do cliente multimídia, sem o emprego de recursos de QoS e com a ocorrência de *delays* pertinentes à *bufferizações* (A) e com o emprego de recursos de QoS e sem a ocorrência de tais *delays* (B).



**Figura 37.** *Frame* de número 563 pertinente ao vídeo de referência original (A) e o *frame* de mesmo número recebido pelo cliente final com o emprego de recursos de QoS (B) e sem o emprego de recursos de QoS (C).



(A)



(B)



(C)

**Figura 38.** PSNR pertinente ao *streaming* de vídeo recebido pelo cliente final sem o emprego de recursos de QoS (linha em vermelho) e com o emprego de recursos de QoS (linha em verde) junto ao *gateway* multimídia.



Nesse contexto, de modo complementar à Figura 38, os valores médios de PSNR obtidos em ambos os casos foram de 38,75 para o *streaming* de vídeo recebido pelo cliente final sem o emprego de recursos de QoS e de 54,27 para o mesmo *streaming* de vídeo recebido pelo cliente final, porém com o emprego de recursos de QoS. Em linhas gerais, esses valores médios classificam o vídeo recebido em ambos os casos como “excelente” considerando a conversão dos valores PSNR para MOS (*Mean Opinion Score*) (GROSS et al., 2004). Ainda assim, é possível observar que o *streaming* de vídeo recebido sem o emprego de recursos de QoS possui grande quantidade de *frames* com valores de PSNR menores que 20 (qualidade “péssima”) e entre 20 e 25 (qualidade “pobre/ruim”), ambos desfavoráveis no âmbito da QoE percebida pelo usuário final. Da mesma forma, também é possível observar que o *streaming* de vídeo recebido com o emprego de recursos de QoS possui grande quantidade de *frames* com valores de PSNR maiores que 37 (qualidade “excelente”), com poucos *frames* de baixa qualidade, ambos favoráveis no âmbito da QoE percebida pelo usuário final.

Nesse cenário, de modo complementar às experimentações descritas anteriormente, duas outras experimentações foram realizadas de modo à melhor exemplificar

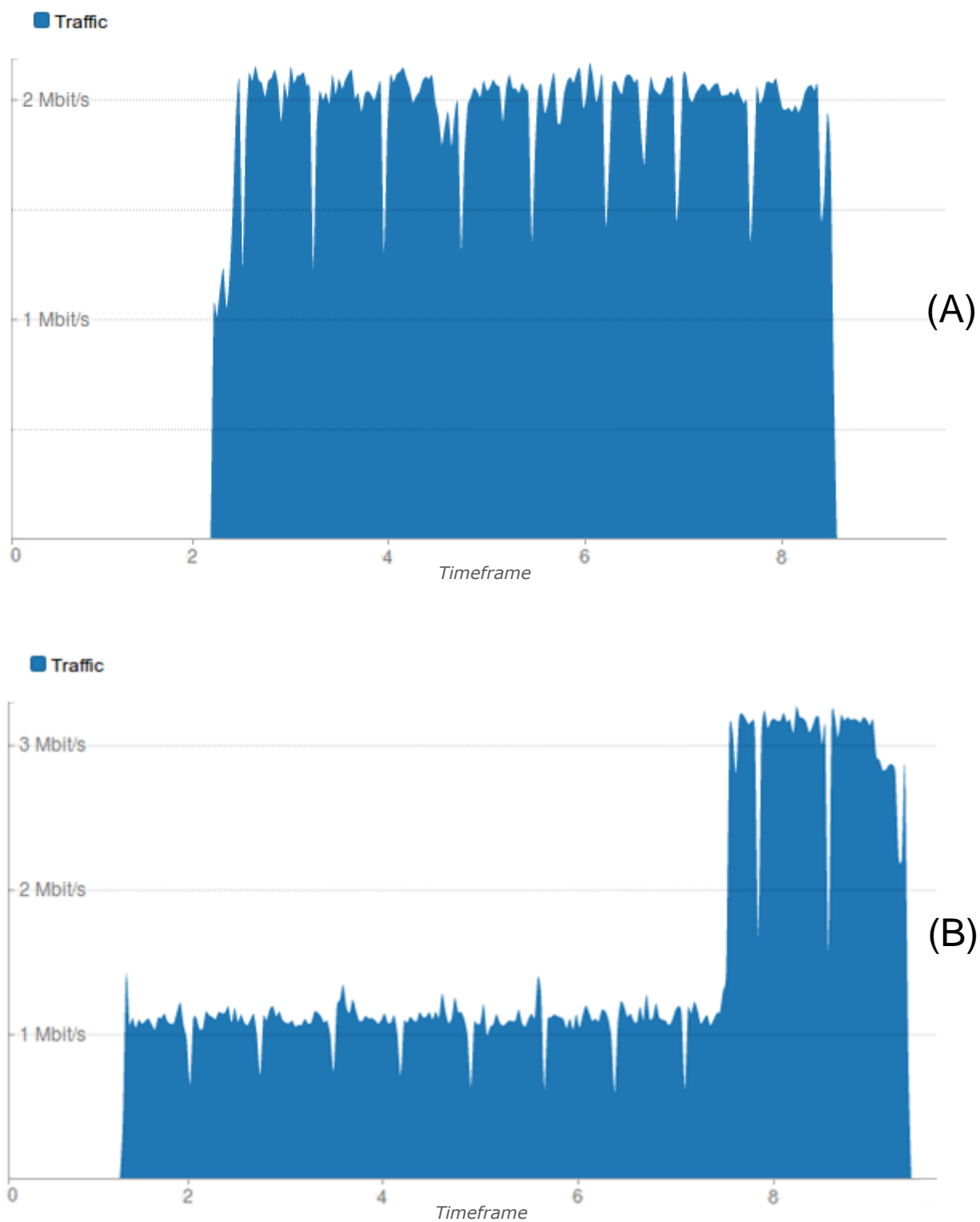
os efeitos da priorização entre as classes de tráfego no âmbito do compartilhamento de recursos de banda de rede entre si:

(I) Transmissões multimídia e simultâneas de dois fluxos de vídeo, com o emprego de recursos de QoS junto ao *gateway* multimídia e com uma classe de tráfego específica para cada fluxo de vídeo em questão. Nesse caso, cada classe de tráfego foi configurada com alocações de banda na ordem de 1 Mbps, podendo compartilhar recursos de banda de rede entre si até 3 Mbps (de acordo com a prioridade de cada classe de tráfego em questão). Para tal, uma das classes de tráfego foi configurada com prioridade 1 (maior prioridade) e a outra classe de tráfego foi configurada com prioridade 2 (menor prioridade). Por sua vez, cada fluxo de vídeo em questão recebeu distintos identificadores DSCP e, por meio de tais identificadores, foram associados às suas respectivas e distintas classes de tráfego junto ao *gateway* multimídia;

(II) Transmissões multimídia e simultâneas de dois fluxos de dados, com o emprego de recursos de QoS junto ao *gateway* multimídia e com uma classe de tráfego específica para cada fluxo de dados em questão. Nesse caso, cada classe de tráfego foi configurada com alocações de banda na ordem de 1 Mbps, podendo compartilhar recursos de banda de rede entre si até 3 Mbps (de acordo com a prioridade de cada classe de tráfego em questão). Para tal, uma das classes de tráfego foi configurada com prioridade 3 (maior prioridade) e a outra classe de tráfego foi configurada com prioridade 4 (menor prioridade). Por sua vez, cada fluxo de dados em questão recebeu distintos identificadores DSCP e, por meio de tais identificadores, foram associados às suas respectivas e distintas classes de tráfego junto ao *gateway* multimídia;

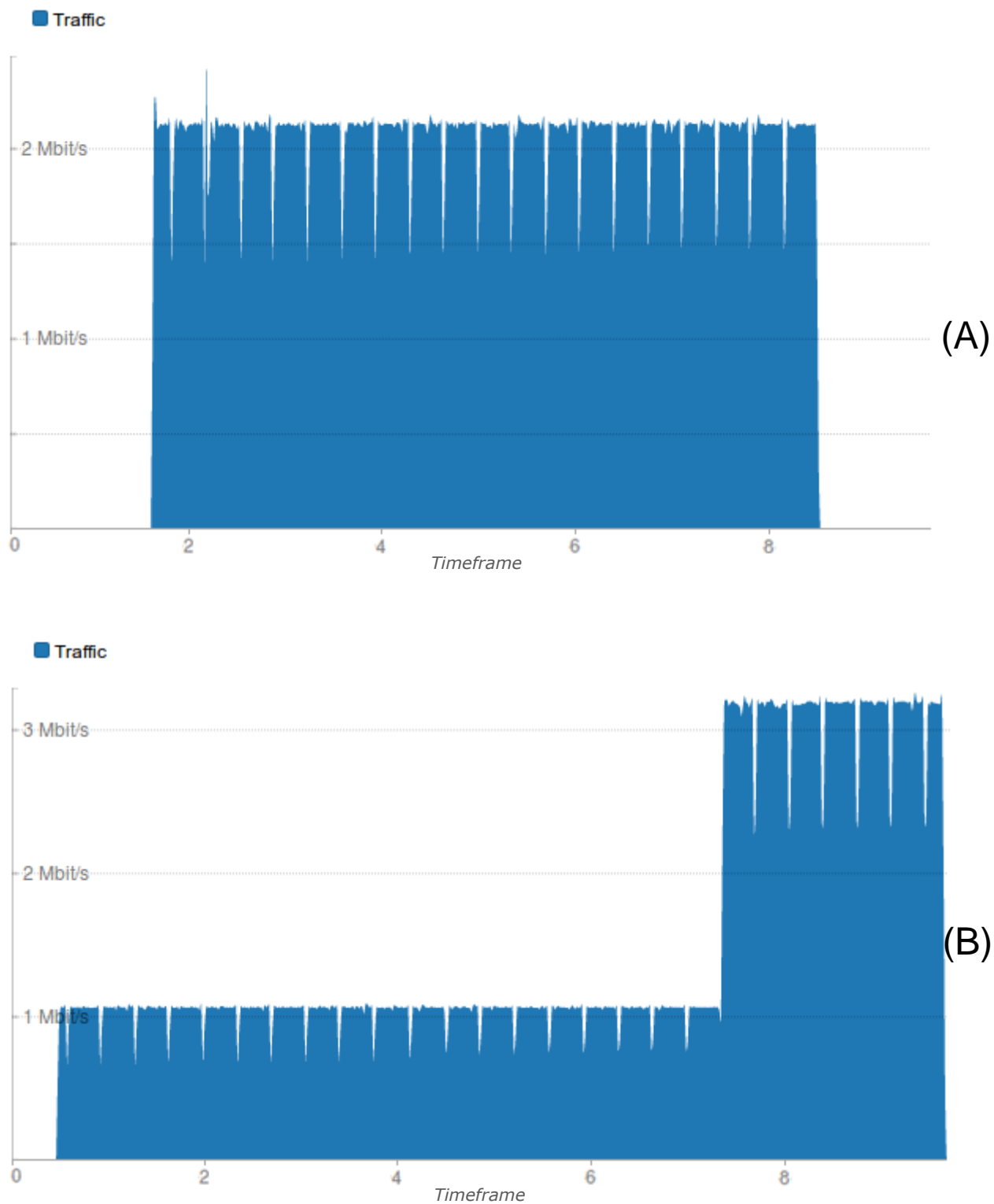
Diante desse cenário, as Figuras 39 e 40 ilustram, respectivamente, a utilização dos recursos de rede na transmissão simultânea dos conteúdos de vídeo entre si e de dados entre si. Por meio de tais ilustrações, observe que os fluxos pertinentes às classes de tráfego de maior prioridade (A) estenderam seus escopos de 1 Mbps para cerca de 2 Mbps ao longo de toda sua transmissão, enquanto os fluxos pertinentes às classes de tráfego de menor prioridade (B) mantiveram seus escopos próximos de 1 Mbps durante tais transmissões, apenas estendendo seus escopos (para cerca de 3 Mbps) após o término de transmissão dos fluxos de maior prioridade.

**Figura 39.** Utilização dos recursos de rede em duas transmissões simultâneas de conteúdos multimídia de vídeo no ambiente de emulação, nesse caso, com o emprego de recursos de QoS junto ao *gateway* multimídia e por meio de classes de tráfego de maior prioridade (A) e de menor prioridade (B).





**Figura 40.** Utilização dos recursos de rede em duas transmissões simultâneas de conteúdos multimídia de dados no ambiente de emulação, nesse caso, com o emprego de recursos de QoS junto ao *gateway* multimídia e por meio de classes de tráfego de maior prioridade (A) e de menor prioridade (B).



Nesse cenário, de modo complementar às Figuras 39 e 40, é importante destacar que ambos os fluxos de tráfego em questão foram transmitidos sobre o protocolo de transporte TCP, via MMS (*Microsoft Media Service*) quanto ao conteúdo de vídeo e via FTP quanto ao conteúdo de dados, ambos utilizando os mesmos arquivos de referência descritos anteriormente. Da mesma forma, é importante destacar que as alocações pertinentes a ambos os conteúdos multimídia em questão foram escolhidas aleatoriamente, porém propositalmente inferiores às demandas exigidas por ambos em suas transmissões individuais (tal como apresentado na Tabela 8), nesse caso, objetivando melhor exemplificar o emprego das prioridades no âmbito dos compartilhamentos de recursos de banda de rede em suas transmissões.

Diante desse cenário, é possível observar que o emprego de recursos de QoS junto ao *gateway* multimídia, além de possibilitar a oferta de diferentes níveis de QoS aos fluxos multimídia experimentados no ambiente de emulação, também possibilitou que fluxos de tráfego distintos, pertinentes ou não ao mesmo conteúdo multimídia, pudessem receber diferentes configurações em termos de prioridades entre si. Em linhas gerais, essa abordagem contribuiu para que a percepção de QoE do usuário final fosse aprimorada, em especial, no âmbito de suas aplicações multimídia de fluxo de tráfego *stream* experimentadas no ambiente de emulação.

#### **4.2.5 – Cache de Conteúdos Multimídia**

Para avaliar o emprego de recursos de cache durante o processo de distribuição de conteúdos multimídia, o *gateway* multimídia foi configurado para atuar como proxy em seu ambiente de rede local. Dessa forma, além de intermediar as requisições realizadas pelos clientes da rede, pode centralizá-las e fornecer recursos que objetivem aprimorar a percepção de QoE do usuário final quanto suas aplicações multimídia, tais como por meio de melhores tempos e taxas de transmissão nos acessos aos conteúdos armazenados e recuperados do cache durante o processo de distribuição dos mesmos ao longo da rede.

Nesse contexto, de modo similar à Seção anterior, três conteúdos multimídia foram considerados na experimentação e avaliação quanto ao emprego de recursos de cache durante o processo de distribuição multimídia na rede local: um conteúdo de áudio, um conteúdo de vídeo e um conteúdo de dados, em ambos os casos, empregando os mesmos

arquivos de referência utilizados anteriormente (Seção 4.2.4). Na rede, esses conteúdos foram disponibilizados a partir do servidor multimídia H1, ambos diretamente sobre um serviço de hospedagem *web* implementado em tal servidor, e, por sua vez, acessados por meio dos clientes multimídia H3 (conteúdo de vídeo), H4 (conteúdo de dados) e H5 (conteúdo de áudio).

Diante desse cenário, quatro experimentações foram realizadas no ambiente de emulação:

- (I) Transmissões multimídia de áudio, de vídeo e de dados, sem o emprego de recursos de cache junto ao *gateway* multimídia, e com ambos os conteúdos sendo acessados individualmente na rede;
- (II) Transmissões multimídia de áudio, de vídeo e de dados, sem o emprego de recursos de cache junto ao *gateway* multimídia, e com ambos os conteúdos sendo acessados simultaneamente na rede;
- (III) Transmissões multimídia de áudio, de vídeo e de dados, com o emprego de recursos de cache junto ao *gateway* multimídia, e com ambos os conteúdos sendo acessados individualmente na rede;
- (IV) Transmissões multimídia de áudio, de vídeo e de dados, com o emprego de recursos de cache junto ao *gateway* multimídia, e com ambos os conteúdos sendo acessados simultaneamente na rede.

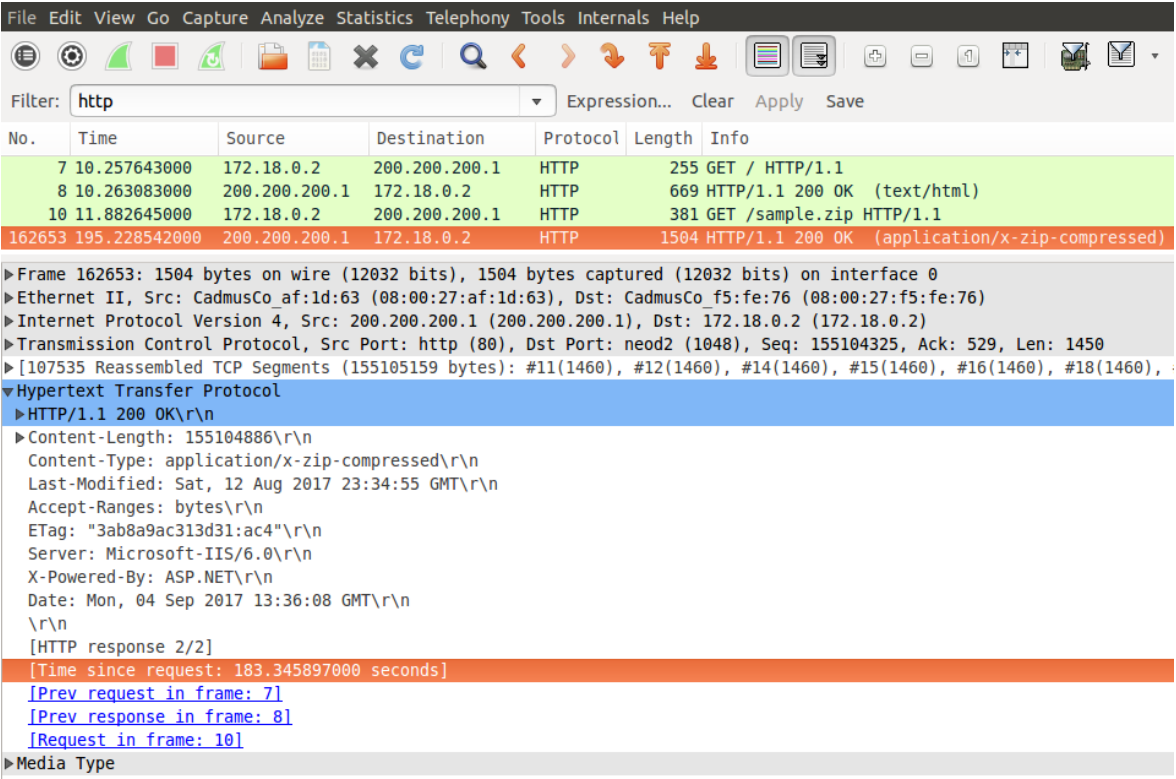
Nesse cenário, para cada experimentação, 5 transmissões de áudio, de vídeo e de dados foram consideradas (transmissões de T1 à T5), ambas realizadas por seu respectivo cliente multimídia, sobre HTTP e, em ambos os casos, utilizando os mesmos arquivos de referência. Em tal cenário, em um primeiro momento, nenhuma configuração no âmbito de recursos de cache foi empregada na rede local. Por sua vez, cada conteúdo multimídia em questão foi acessado individualmente na rede. Em seguida, ainda sem o emprego de recursos de cache, ambos os conteúdos multimídia foram acessados simultaneamente, nesse caso, com cada conteúdo em específico sendo requisitado por seu respectivo cliente multimídia. Ao final, empregando recursos de cache junto ao *gateway* multimídia, repetiu-se a mesma estratégia adotada anteriormente, com ambos os conteúdos multimídia sendo acessados individualmente em um primeiro momento e simultaneamente em um segundo momento.

Diante desse cenário, com o objetivo de avaliar, em especial, os tempos e taxas de transmissão para ambos os conteúdos multimídia experimentais, nesse caso, sem o emprego de recursos de cache e com o emprego de recursos de cache junto ao *gateway* multimídia, observações quanto a tais tempos e taxas de transmissão foram realizadas no ambiente de emulação. Para tal, no âmbito dos tempos de transmissão, as observações foram realizadas por meio das requisições HTTP pertinentes a cada conteúdo multimídia de áudio, de vídeo e de dados em questão. Em linhas gerais, essas observações tiveram como base os tempos medidos entre as mensagens HTTP de requisição e de entrega de ambos os conteúdos multimídia na rede (obtidos por meio do analisador de protocolos Wireshark). De modo complementar, esses tempos de transmissão também foram observados (subjetivamente) por meio das aplicações multimídia de áudio, de vídeo e de dados empregadas nos clientes multimídia da rede local.

Nesse contexto, como exemplo, as Figuras 41, 42 e 43 ilustram observações sobre um dos tempos de transmissão para os conteúdos multimídia de dados, de vídeo e de áudio, respectivamente, no ambiente de emulação. Em ambos os casos, as ilustrações consideram observações tanto no âmbito da rede (A), quanto no âmbito da aplicação multimídia (B). Por meio de tais ilustrações, observe que, no âmbito da rede, os tempos de transmissão observados para ambos os conteúdos multimídia foram de cerca de 180 segundos (especificamente, 183,345897000 segundos) para o conteúdo de dados, de cerca de 90 segundos (especificamente 87,710512000 segundos) para o conteúdo de vídeo e de cerca de 40 segundos (especificamente, 38,216989000 segundos) para o conteúdo de áudio, em ambos os casos, obtidos por meio do analisador de protocolos Wireshark. De modo complementar, no âmbito das aplicações multimídia, esses tempos de transmissão também puderam ser acompanhados (subjetivamente) por meio de informações disponibilizadas por ambas em seus respectivos clientes multimídia. Para tal, como exemplo, o painel inferior (B) da Figura 41 ilustra, em especial, o tempo de transmissão pertinente ao conteúdo de dados junto à mensagem indicativa de conclusão de *download* fornecida pelo próprio *browser* do cliente multimídia H4. Por sua vez, o painel inferior (B) das Figuras 42 e 43 ilustram, em especial, informações de *status* de *download* (exibidas nas barras de *status* e de progresso dos *players* de vídeo e de áudio), bem como de estatísticas de transmissão (em especial, quanto às taxas de transmissão empregadas naquele momento em específico) quanto aos conteúdos multimídia de vídeo e de áudio, respectivamente, junto aos próprios *players* de vídeo e de áudio dos clientes multimídia H3 e H5.

**Figura 41.** Exemplo de um dos tempos de transmissão para o conteúdo multimídia de dados sobre HTTP: observações realizadas no âmbito da rede (A) e no âmbito da aplicação multimídia (B).

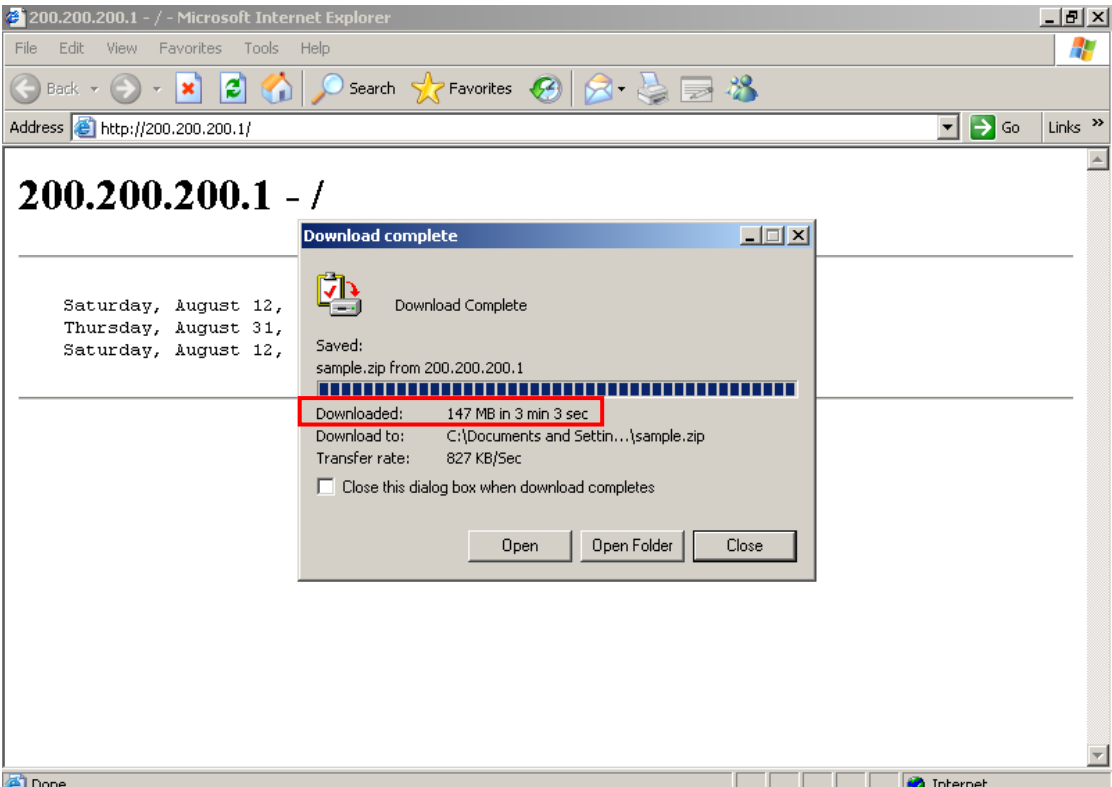
(A)



The screenshot shows a Wireshark capture of an HTTP transaction. The packet list pane shows three packets: a GET request (No. 7), an HTML response (No. 8), and a compressed file response (No. 10). Packet 10 is selected, and the packet details pane shows the following information:

- Frame 162653: 1504 bytes on wire (12032 bits), 1504 bytes captured (12032 bits) on interface 0
- Ethernet II, Src: CadmusCo\_af:1d:63 (08:00:27:af:1d:63), Dst: CadmusCo\_f5:fe:76 (08:00:27:f5:fe:76)
- Internet Protocol Version 4, Src: 200.200.200.1 (200.200.200.1), Dst: 172.18.0.2 (172.18.0.2)
- Transmission Control Protocol, Src Port: http (80), Dst Port: neod2 (1048), Seq: 155104325, Ack: 529, Len: 1450
- [107535 Reassembled TCP Segments (155105159 bytes): #11(1460), #12(1460), #14(1460), #15(1460), #16(1460), #18(1460), ...]
- Hypertext Transfer Protocol
  - HTTP/1.1 200 OK\r\n
  - Content-Length: 155104886\r\n
  - Content-Type: application/x-zip-compressed\r\n
  - Last-Modified: Sat, 12 Aug 2017 23:34:55 GMT\r\n
  - Accept-Ranges: bytes\r\n
  - ETag: "3ab8a9ac313d31:ac4"\r\n
  - Server: Microsoft-IIS/6.0\r\n
  - X-Powered-By: ASP.NET\r\n
  - Date: Mon, 04 Sep 2017 13:36:08 GMT\r\n
  - \r\n
  - [HTTP response 2/2]
- [Time since request: 183.345897000 seconds]
- [Prev request in frame: 7]
- [Prev response in frame: 8]
- [Request in frame: 10]
- Media Type

(B)

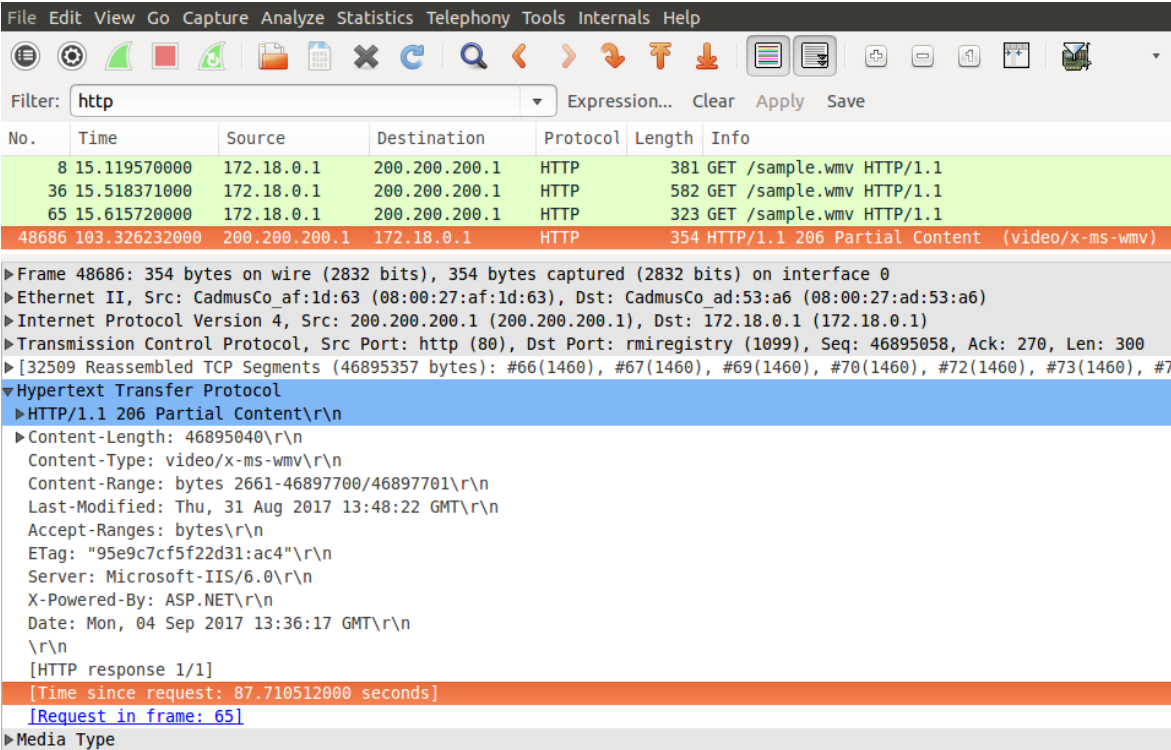


The screenshot shows Microsoft Internet Explorer displaying the address `http://200.200.200.1/`. A "Download complete" dialog box is open, showing the following information:

- Download Complete
- Saved: sample.zip from 200.200.200.1
- Downloaded: 147 MB in 3 min 3 sec
- Download to: C:\Documents and Settings\...\sample.zip
- Transfer rate: 827 KB/Sec
- Close this dialog box when download completes

Buttons: Open, Open Folder, Close

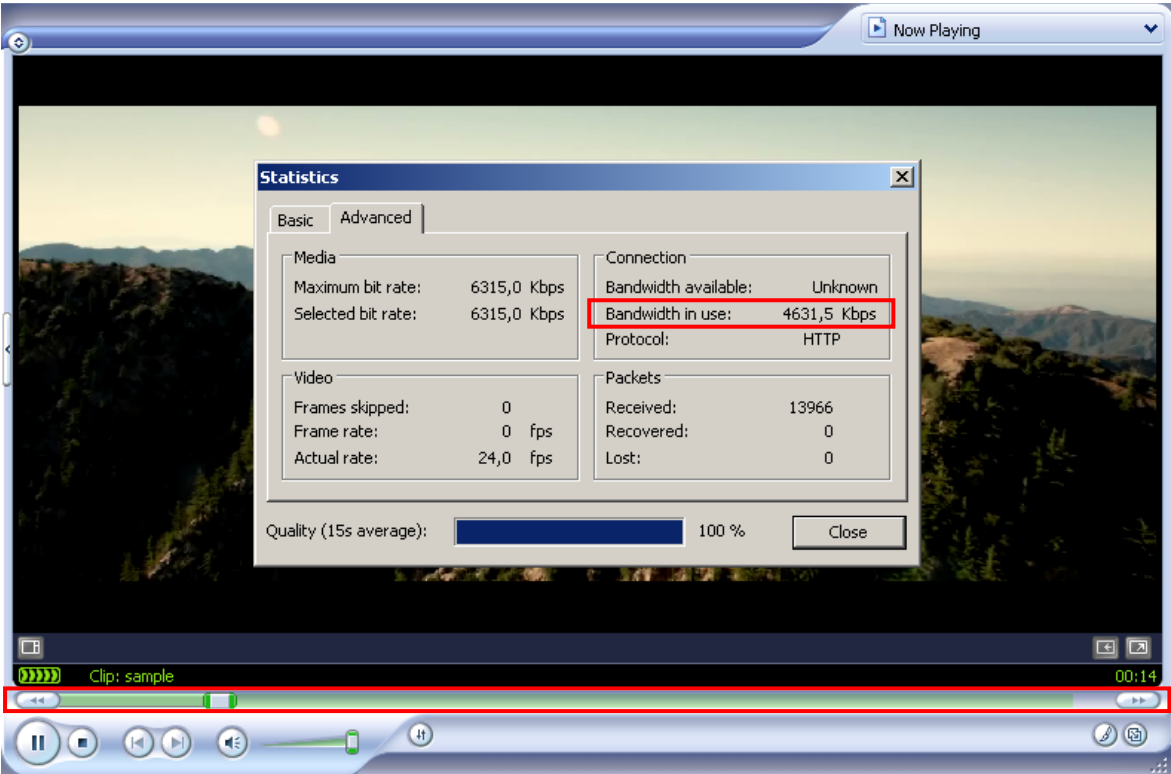
**Figura 42.** Exemplo de um dos tempos de transmissão para o conteúdo multimídia de vídeo sobre HTTP: observações realizadas no âmbito da rede (A) e no âmbito da aplicação multimídia (B).



(A)

No.	Time	Source	Destination	Protocol	Length	Info
8	15.119570000	172.18.0.1	200.200.200.1	HTTP	381	GET /sample.wmv HTTP/1.1
36	15.518371000	172.18.0.1	200.200.200.1	HTTP	582	GET /sample.wmv HTTP/1.1
65	15.615720000	172.18.0.1	200.200.200.1	HTTP	323	GET /sample.wmv HTTP/1.1
48686	103.326232000	200.200.200.1	172.18.0.1	HTTP	354	HTTP/1.1 206 Partial Content (video/x-ms-wmv)

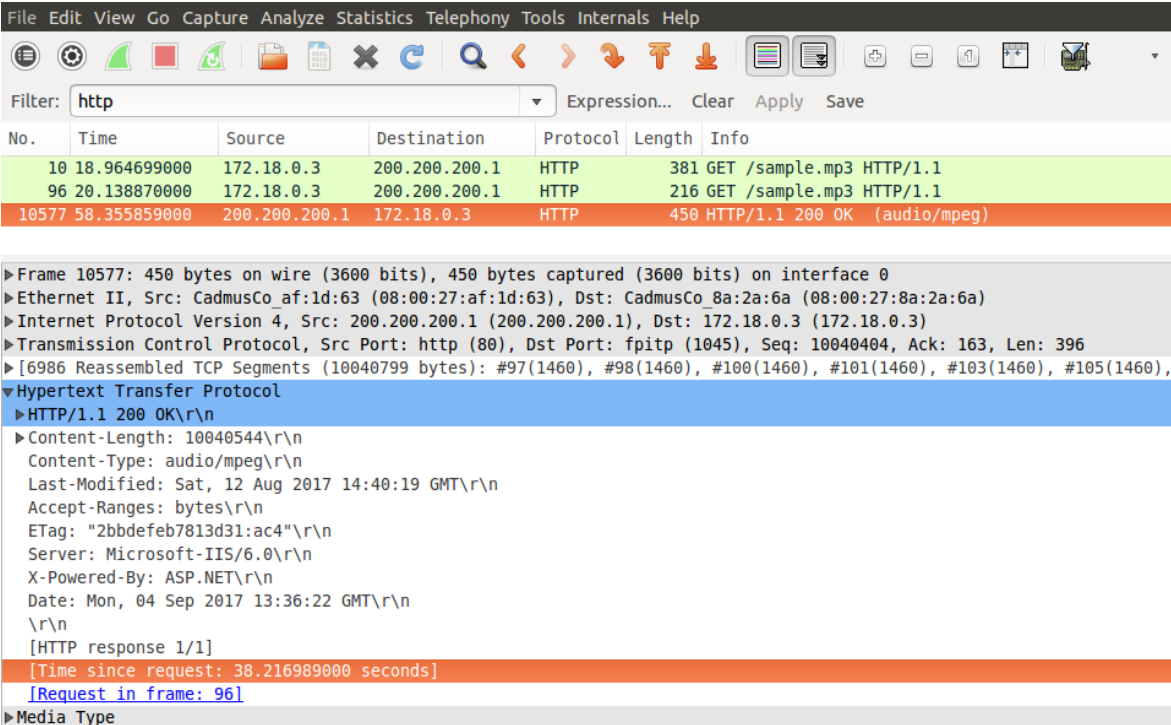
▶Frame 48686: 354 bytes on wire (2832 bits), 354 bytes captured (2832 bits) on interface 0  
 ▶Ethernet II, Src: CadmusCo\_af:1d:63 (08:00:27:af:1d:63), Dst: CadmusCo\_ad:53:a6 (08:00:27:ad:53:a6)  
 ▶Internet Protocol Version 4, Src: 200.200.200.1 (200.200.200.1), Dst: 172.18.0.1 (172.18.0.1)  
 ▶Transmission Control Protocol, Src Port: http (80), Dst Port: rmiregistry (1099), Seq: 46895058, Ack: 270, Len: 300  
 ▶[32509 Reassembled TCP Segments (46895357 bytes): #66(1460), #67(1460), #69(1460), #70(1460), #72(1460), #73(1460), #7  
 ▶Hypertext Transfer Protocol  
 ▶HTTP/1.1 206 Partial Content\r\n
 ▶Content-Length: 46895040\r\n
 Content-Type: video/x-ms-wmv\r\n
 Content-Range: bytes 2661-46897700/46897701\r\n
 Last-Modified: Thu, 31 Aug 2017 13:48:22 GMT\r\n
 Accept-Ranges: bytes\r\n
 ETag: "95e9c7cf5f22d31:ac4"\r\n
 Server: Microsoft-IIS/6.0\r\n
 X-Powered-By: ASP.NET\r\n
 Date: Mon, 04 Sep 2017 13:36:17 GMT\r\n
 \r\n
 [HTTP response 1/1]  
 [Time since request: 87.710512000 seconds]  
 [Request in frame: 65]  
 ▶Media Type



(B)

Category	Field	Value
Media	Maximum bit rate:	6315,0 Kbps
	Selected bit rate:	6315,0 Kbps
Connection	Bandwidth available:	Unknown
	Bandwidth in use:	4631,5 Kbps
Packets	Received:	13966
	Recovered:	0
	Lost:	0
Video	Frames skipped:	0
Video	Frame rate:	0 fps
Video	Actual rate:	24,0 fps
Quality (15s average):		100 %

**Figura 43.** Exemplo de um dos tempos de transmissão para o conteúdo multimídia de áudio sobre HTTP: observações realizadas no âmbito da rede (A) e no âmbito da aplicação multimídia (B).



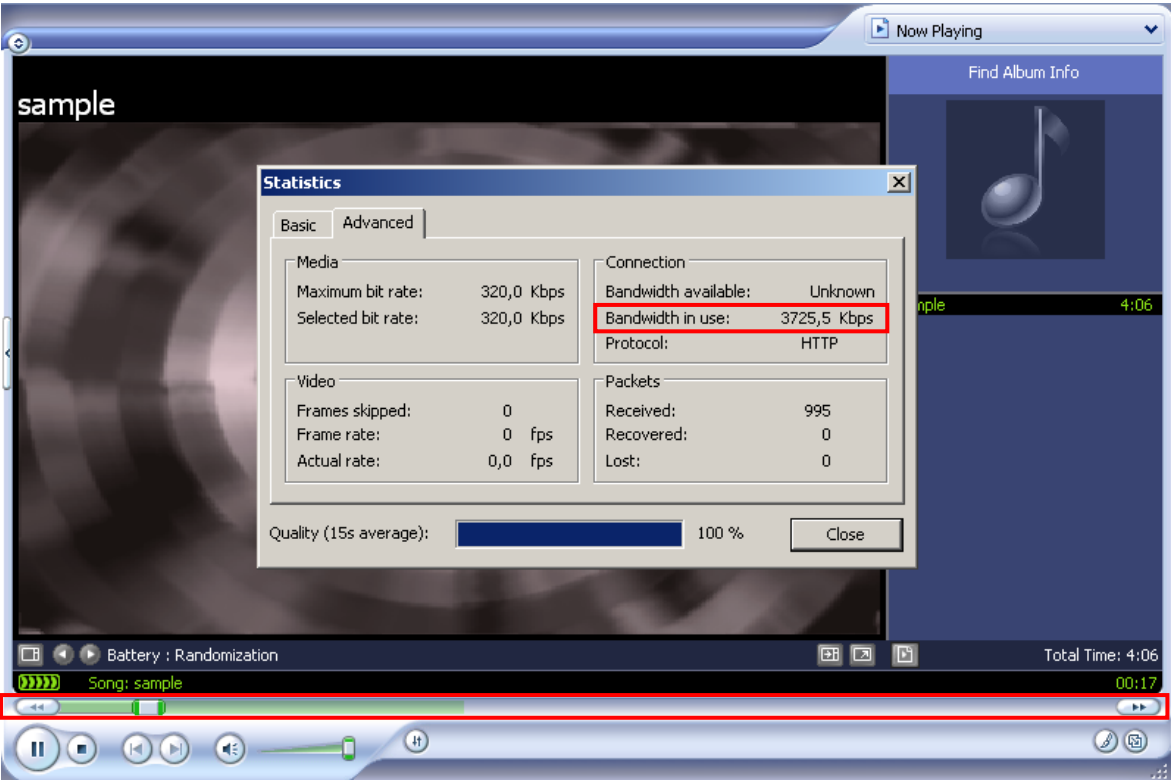
The screenshot shows the Wireshark interface with a filter set to 'http'. The packet list pane displays three packets:

No.	Time	Source	Destination	Protocol	Length	Info
10	18.964699000	172.18.0.3	200.200.200.1	HTTP	381	GET /sample.mp3 HTTP/1.1
96	20.138870000	172.18.0.3	200.200.200.1	HTTP	216	GET /sample.mp3 HTTP/1.1
10577	58.355859000	200.200.200.1	172.18.0.3	HTTP	450	HTTP/1.1 200 OK (audio/mpeg)

The packet details pane for packet 10577 shows the following structure:

- Frame 10577: 450 bytes on wire (3600 bits), 450 bytes captured (3600 bits) on interface 0
- Ethernet II, Src: CadmusCo\_af:1d:63 (08:00:27:af:1d:63), Dst: CadmusCo\_8a:2a:6a (08:00:27:8a:2a:6a)
- Internet Protocol Version 4, Src: 200.200.200.1 (200.200.200.1), Dst: 172.18.0.3 (172.18.0.3)
- Transmission Control Protocol, Src Port: http (80), Dst Port: fptip (1045), Seq: 10040404, Ack: 163, Len: 396
- [6986 Reassembled TCP Segments (10040799 bytes): #97(1460), #98(1460), #100(1460), #101(1460), #103(1460), #105(1460),
- Hypertext Transfer Protocol
  - HTTP/1.1 200 OK\r\n
  - Content-Length: 10040544\r\n
  - Content-Type: audio/mpeg\r\n
  - Last-Modified: Sat, 12 Aug 2017 14:40:19 GMT\r\n
  - Accept-Ranges: bytes\r\n
  - ETag: "2bbdefeb7813d31:ac4"\r\n
  - Server: Microsoft-IIS/6.0\r\n
  - X-Powered-By: ASP.NET\r\n
  - Date: Mon, 04 Sep 2017 13:36:22 GMT\r\n
  - \r\n
  - [HTTP response 1/1]

A red arrow points to the line: [Time since request: 38.216989000 seconds]



The screenshot shows a media player interface with a 'Statistics' window open. The 'Basic' tab is selected, displaying the following information:

Media		Connection	
Maximum bit rate:	320,0 Kbps	Bandwidth available:	Unknown
Selected bit rate:	320,0 Kbps	Bandwidth in use:	3725,5 Kbps
		Protocol:	HTTP
Video		Packets	
Frames skipped:	0	Received:	995
Frame rate:	0 fps	Recovered:	0
Actual rate:	0,0 fps	Lost:	0

The 'Quality (15s average)' is shown as 100%. The media player interface also shows the song 'sample' and a progress bar at the bottom.

Diante desse cenário, a Tabela 10 apresenta os tempos de transmissão observados para as 5 transmissões (de T1 à T5) pertinentes aos conteúdos multimídia de áudio, de vídeo e de dados, em transmissões individuais, sem o emprego de recursos de cache e com o emprego de recursos de cache junto ao *gateway* multimídia. De modo complementar, a Tabela 11 apresenta os tempos de transmissão observados para as 5 transmissões pertinentes aos mesmos conteúdos multimídia em questão, em transmissões simultâneas, sem o emprego de recursos de cache e com o emprego de recursos de cache junto ao *gateway* multimídia.

**Tabela 10.** Tempos de transmissão para os conteúdos multimídia de áudio, de vídeo e de dados, em transmissões individuais, sem o emprego de recursos de cache e com o emprego de recursos de cache junto ao *gateway* multimídia.

Tempos de transmissão para os conteúdos multimídia de áudio, de vídeo e de dados (em transmissões individuais)												
Conteúdo de	Sem recursos de cache						Com recursos de cache					
	T1	T2	T3	T4	T5	Média	T1	T2	T3	T4	T5	Média
<b>Áudio</b>	27309 ms	27175 ms	27232 ms	27379 ms	27241 ms	<b>27267</b> <b>ms</b>	27110 ms	717 ms	612 ms	543 ms	698 ms	<b>5936</b> <b>ms</b>
<b>Vídeo</b>	39412 ms	40398 ms	39572 ms	40730 ms	40466 ms	<b>40044</b> <b>ms</b>	39440 ms	11484 ms	10438 ms	11584 ms	11861 ms	<b>16961</b> <b>ms</b>
<b>Dados</b>	134992 ms	133496 ms	136704 ms	137195 ms	137820 ms	<b>136041</b> <b>ms</b>	136086 ms	34561 ms	33942 ms	31908 ms	39382 ms	<b>55176</b> <b>ms</b>

**Tabela 11.** Tempos de transmissão para os conteúdos multimídia de áudio, de vídeo e de dados, em transmissões simultâneas, sem o emprego de recursos de cache e com o emprego de recursos de cache junto ao *gateway* multimídia.

Tempos de transmissão para os conteúdos multimídia de áudio, de vídeo e de dados (em transmissões simultâneas)												
Conteúdo de	Sem recursos de cache						Com recursos de cache					
	T1	T2	T3	T4	T5	Média	T1	T2	T3	T4	T5	Média
<b>Áudio</b>	38217 ms	39124 ms	35825 ms	38455 ms	40086 ms	<b>38341</b> <b>ms</b>	26887 ms	9940 ms	13642 ms	16207 ms	9691 ms	<b>15273</b> <b>ms</b>
<b>Vídeo</b>	87711 ms	87793 ms	87160 ms	87914 ms	88270 ms	<b>87770</b> <b>ms</b>	88389 ms	14369 ms	14165 ms	13944 ms	13928 ms	<b>28959</b> <b>ms</b>
<b>Dados</b>	183346 ms	192216 ms	185062 ms	187872 ms	190344 ms	<b>187768</b> <b>ms</b>	184981 ms	41092 ms	38502 ms	36738 ms	37080 ms	<b>67679</b> <b>ms</b>

Diante dos resultados obtidos, é possível observar que, individualmente e sem o emprego de recursos de cache junto ao *gateway* multimídia, os tempos médios de transmissão observados para as 5 transmissões pertinentes aos conteúdos multimídia de áudio, de vídeo e



de dados foram de, respectivamente, 27267 ms, 40044 ms e 136041 ms. Por outro lado, no acesso simultâneo aos mesmos conteúdos multimídia na rede, seus tempos de transmissão foram superiores. Nesse caso, os tempos médios de transmissão observados para ambos os conteúdos multimídia foram de 38341 ms (conteúdo de áudio), 87770 ms (conteúdo de vídeo) e 187768 ms (conteúdo de dados). Em linhas gerais, esse aumento nos tempos de transmissão pode afetar negativamente a percepção de QoE do usuário final, em especial, quanto às aplicações pertinentes aos *streamings* de áudio e de vídeo, tais como por meio de eventuais *delays* pertinentes à *bufferizações* junto aos *players* dos clientes multimídia ou por meio de eventuais degradações de qualidade de áudio e/ou vídeo pertinentes às baixas taxas de transmissão empregadas durante a reprodução de ambos, de modo geral.

Por outro lado, ainda por meio das Tabelas 10 e 11, é possível observar que os tempos médios de transmissão para os mesmos conteúdos multimídia foram melhores (menores) a partir do emprego de recursos de cache junto ao *gateway* multimídia. Nesse caso, é possível observar que, individualmente, tais tempos médios de transmissão foram de 5936 ms para o conteúdo de áudio, 16961 ms para conteúdo de vídeo e 55176 ms para o conteúdo de dados. Da mesma forma, no acesso simultâneo aos mesmos conteúdos multimídia, seus tempos médios de transmissão também foram melhores (menores), nesse caso, na ordem de 15273 ms para o conteúdo de áudio, de 28959 ms para o conteúdo de vídeo e 67679 ms para o conteúdo de dados. Em ambos os casos (transmissões individuais e simultâneas com o emprego de recursos de cache junto ao *gateway* multimídia), é possível observar que as reduções nos tempos de transmissão dos conteúdos de áudio, de vídeo e de dados foram especialmente melhores (menores) a partir do segundo acesso (transmissão) ao conteúdo multimídia em questão. Em linhas gerais, isso se deve ao fato de que na primeira requisição ao conteúdo multimídia, o mesmo ainda não estava armazenado em cache junto ao *gateway* multimídia, sendo recuperado da rede (servidor multimídia) pelo *gateway* multimídia para, então, ser distribuído aos clientes da rede local. Por sua vez, a partir dessa requisição inicial, o conteúdo multimídia em questão foi armazenado em cache junto ao *gateway* multimídia e, na segunda e demais requisições ao mesmo, recuperado e distribuído aos clientes da rede local diretamente por meio de tal recurso de cache.

Nesse contexto, como exemplo e de modo complementar às Tabelas 10 e 11, a Figura 44 ilustra informações sobre a utilização de recursos de cache junto ao *gateway* multimídia no primeiro acesso aos conteúdos multimídia de áudio, de vídeo e de dados no ambiente de emulação.




**Figura 44.** Utilização de recursos de cache junto ao *gateway* multimídia no primeiro acesso aos conteúdos multimídia de áudio, de vídeo e de dados no ambiente de emulação.

NUM	USERID	CONNECT	BYTES	%BYTES	IN-CACHE-OUT
1	172.18.0.2	1	155.10M	73,15%	0,00% 100,00%
2	172.18.0.1	1	46.89M	22,12%	0,00% 100,00%
3	172.18.0.3	1	10.04M	4,74%	0,00% 100,00%
<b>TOTAL</b>		<b>3</b>	<b>212.04M</b>		<b>0,00% 100,00%</b>
<b>AVERAGE</b>		<b>1</b>	<b>70.68M</b>		




Por meio de tal ilustração, é possível observar por meio da coluna dupla “*IN-CACHE-OUT*”, que todas as requisições realizadas pelos clientes multimídia até o momento (nesse caso, uma única requisição para cada conteúdo multimídia em questão, observada por meio da coluna “*CONNECT*”) tiveram seus conteúdos recuperados de modo externo ao cache (100% de “*CACHE-OUT*”), ou seja, por meio da rede/Internet. Nesse caso, as requisições realizadas pelo cliente multimídia H4 (*host* 172.18.0.2), pertinentes ao conteúdo multimídia de dados, em um volume total de 155,10 MB (observada por meio da coluna “*BYTES*”) representaram, em termos de volume de tráfego, 73,15% de todo o volume trafegado até o momento; as requisições realizadas pelo cliente multimídia H3 (*host* 172.18.0.1), pertinentes ao conteúdo multimídia de vídeo, em um volume total de 46,89 MB representaram, em termos de volume de tráfego, 22,12% de todo o volume trafegado até o momento; e, por sua vez, as requisições realizadas pelo cliente multimídia H5 (*host* 172.18.0.3), pertinentes ao conteúdo multimídia de áudio, em um volume total de 10,04 MB representaram, em termos de volume de tráfego, 4,74% de todo o volume trafegado até o momento.

Por sua vez, por meio da Figura 45, é possível observar informações sobre a utilização de recursos de cache junto ao *gateway* multimídia no segundo acesso (A) e no quinto acesso (B) aos mesmos conteúdos multimídia de áudio, de vídeo e de dados no ambiente de emulação. Nesse caso, observe, por meio da coluna dupla “*IN-CACHE-OUT*”, que na segunda requisição aos conteúdos multimídia em questão, o recurso de cache foi empregado em 50% das mesmas (ou seja, em 100% da segunda transmissão experimental, T2) e, por sua vez, que na quinta requisição aos mesmos conteúdos multimídia em questão, o recurso de cache foi empregado em 80% das mesmas (ou seja, em 100% da segunda até a quinta transmissão experimental, de T2 à T5).

**Figura 45.** Utilização de recursos de cache junto ao *gateway* multimídia no segundo acesso (A) e no quinto acesso (B) aos mesmos conteúdos multimídia de áudio, de vídeo e de dados no ambiente de emulação.

NUM		USERID	CONNECT	BYTES	%BYTES	IN-CACHE-OUT	
1		172.18.0.2	2	310.21M	73,15%	50,00%	50,00%
2		172.18.0.1	2	93.79M	22,12%	50,00%	50,00%
3		172.18.0.3	2	20.08M	4,74%	50,00%	50,00%
<b>TOTAL</b>			<b>6</b>	<b>424.08M</b>		<b>50,00%</b>	<b>50,00%</b>
<b>AVERAGE</b>			<b>2</b>	<b>141.36M</b>			

(A)

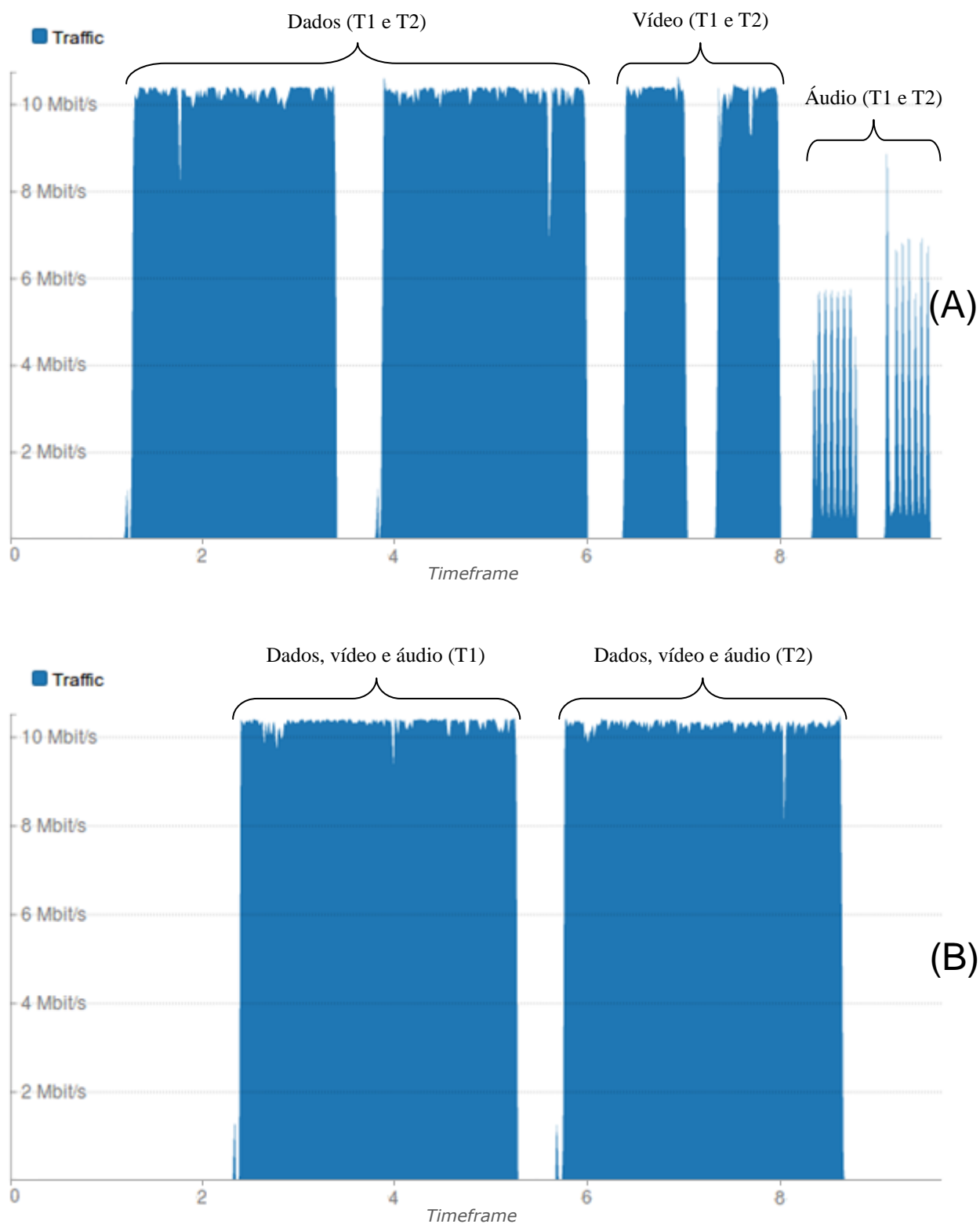
NUM		USERID	CONNECT	BYTES	%BYTES	IN-CACHE-OUT	
1		172.18.0.2	5	775.52M	73,15%	80,00%	20,00%
2		172.18.0.1	5	234.47M	22,12%	80,00%	20,00%
3		172.18.0.3	5	50.19M	4,73%	79,99%	20,01%
<b>TOTAL</b>			<b>15</b>	<b>1.06G</b>		<b>80,00%</b>	<b>20,00%</b>
<b>AVERAGE</b>			<b>5</b>	<b>353.39M</b>			

(B)

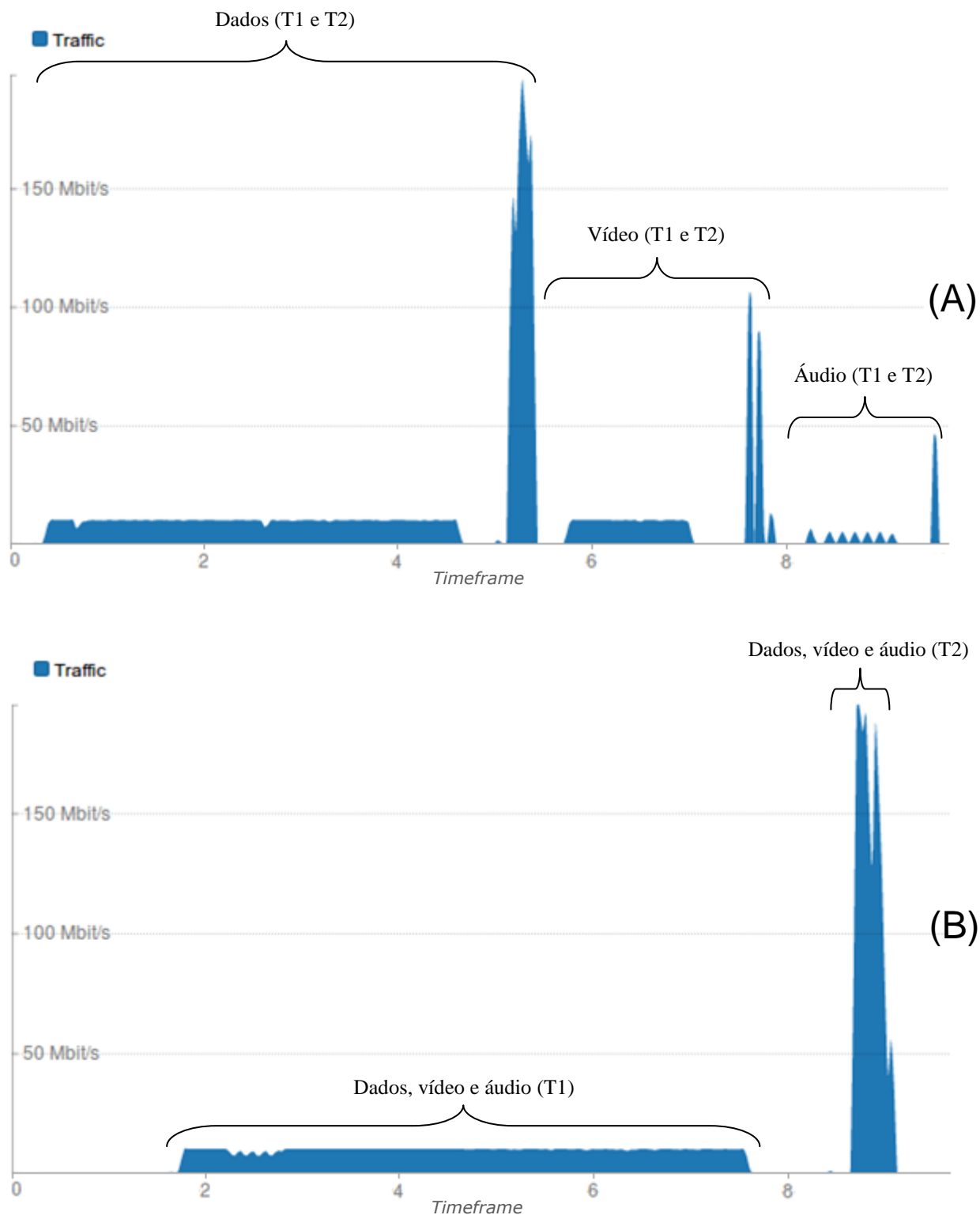
Nesse contexto, de modo complementar a tais ilustrações, considerando que 80% de todo volume de tráfego distribuído aos clientes da rede local, na ordem de 1,06 GB, foram obtidos por meio de recursos de cache empregados internamente na LAN (por meio do *gateway* multimídia), no quinto acesso aos mesmos conteúdos multimídia de áudio, de vídeo e de dados no ambiente de emulação, cerca de 800 MB deixaram de ser obtidos da “Internet” (servidores multimídia na “Internet emulada”) durante esse processo de distribuição de conteúdos multimídia. Em um contexto mais amplo, essa abordagem é de extrema importância para amenizar a utilização dos enlaces de rede entre os clientes multimídia e o ISP na qual o servidor multimídia na Internet está conectado, minimizando, por exemplo, eventuais congestionamentos, atrasos e perdas de pacotes durante esse processo de recuperação e distribuição de conteúdos multimídia, de modo geral.

Além disso, as taxas de transmissão observadas para os conteúdos multimídia em questão foram superiores (melhores) quando ambos foram recuperados e distribuídos por meio do recurso de cache junto ao *gateway* multimídia na rede local. Como exemplo, sem o emprego de recursos de cache, as taxas máximas de transmissão observadas foram de cerca de 10 Mbps (taxa máxima de transmissão pertinente ao “*link*” de Internet empregado no ambiente de emulação), tanto no acesso individual, quanto no acesso simultâneo aos conteúdos multimídia em questão (Figura 46). Por outro lado, com o emprego de recursos de cache, essas taxas só foram observadas no primeiro acesso aos conteúdos multimídia em questão, ultrapassando 150 Mbps no segundo e demais acessos (individuais e, também, simultâneos) a tais conteúdos multimídia (Figura 47).

**Figura 46.** Taxas de transmissão observadas em duas transmissões individuais (A) e em duas transmissões simultâneas (B), tidas como T1 e T2, quanto aos conteúdos multimídia de dados, de vídeo e de áudio, respectivamente, no ambiente de emulação, ambas sem o emprego de recursos de cache junto ao *gateway* multimídia.



**Figura 47.** Taxas de transmissão observadas nas duas primeiras transmissões individuais (A) e nas duas primeiras transmissões simultâneas (B), tidas como T1 e T2, quanto aos conteúdos multimídia de dados, de vídeo e de áudio, respectivamente, no ambiente de emulação, ambas com o emprego de recursos de cache junto ao *gateway* multimídia.



Nesse cenário, o emprego de recursos de cache também é importante para minimizar eventuais atrasos e/ou perdas de pacotes, provenientes no âmbito Internet, durante o processo de distribuição de conteúdos multimídia aos clientes da rede local (nesse caso, referindo-se aos conteúdos multimídia já armazenados em cache e que não necessitam ser recuperados de seus respectivos servidores multimídia na Internet).

Nesse contexto, de modo complementar às experimentações descritas anteriormente, duas outras experimentações foram realizadas de modo à exemplificar os tempos e taxas de transmissão na distribuição de conteúdos multimídia em um cenário com elevado percentual de perda de pacotes e em um cenário com elevado *delay* de transmissão, ambos no âmbito “Internet” pertinente ao ambiente de emulação. Para tal, por meio da ferramenta netem<sup>38</sup>, na interface “Internet” do roteador R1, que corresponde ao enlace Internet fornecido ao *gateway* multimídia no ambiente de emulação, considerou-se, quanto as perda de pacotes, o valor de 10% e, por sua vez, quanto aos *delays* de transmissão, o valor de 200 ms, ambos aplicados para todo fluxo de tráfego transmitido por meio de tal interface e em um experimento específico para tal.

Diante desse cenário, como exemplo, as Figuras 48 e 49 ilustram, por meio de mensagens ICMP (via “ping”) entre o cliente multimídia H3 e o servidor multimídia H1, os efeitos de tais configurações no âmbito de percentuais de pacotes perdidos e de *delays* de transmissão, respectivamente, no ambiente de emulação. Por sua vez, quanto à distribuição de conteúdos multimídia em tais cenários, no âmbito da experimentação envolvendo perda de pacotes, as Figuras 50 e 51 ilustram, respectivamente, os tempos e taxas de transmissão pertinentes a dois acessos ao conteúdo multimídia de vídeo, nesse caso, sem o emprego de recursos de cache na rede local. De modo similar, as Figuras 52 e 53 ilustram, respectivamente, os tempos e taxas de transmissão pertinentes aos dois primeiros acessos ao mesmo conteúdo multimídia de vídeo em questão, porém com o emprego de recursos de cache junto *gateway* multimídia na rede local. Por sua vez, no âmbito da experimentação envolvendo *delays* de transmissão, as Figuras 54 e 55 ilustram, respectivamente, os tempos e taxas de transmissão pertinentes a dois acessos ao conteúdo multimídia de vídeo, nesse caso, sem o emprego de recursos de cache na rede local. De modo similar, as Figuras 56 e 57 ilustram, respectivamente, os tempos e taxas de transmissão pertinentes aos dois primeiros acessos ao mesmo conteúdo multimídia de vídeo em questão, porém com o emprego de recursos de cache junto *gateway* multimídia na rede local.

---

<sup>38</sup> netem em <https://wiki.linuxfoundation.org/networking/netem>

**Figura 48.** Mensagens ICMP (via “ping”) entre o cliente multimídia H3 e o servidor multimídia H1 sem a inserção de perdas de pacotes (A) e com a inserção de perdas de pacotes na ordem de 10% (B) na interface “Internet” do roteador R1.

```
C:\Documents and Settings\aluno>ping -n 10 h1.diorio.corp.br
Pinging h1.diorio.corp.br [200.200.200.1] with 32 bytes of data:
Reply from 200.200.200.1: bytes=32 time=4ms TTL=126
Reply from 200.200.200.1: bytes=32 time=2ms TTL=126
Reply from 200.200.200.1: bytes=32 time<1ms TTL=126
Reply from 200.200.200.1: bytes=32 time=1ms TTL=126
Reply from 200.200.200.1: bytes=32 time=4ms TTL=126
Reply from 200.200.200.1: bytes=32 time=1ms TTL=126
Reply from 200.200.200.1: bytes=32 time=4ms TTL=126
Reply from 200.200.200.1: bytes=32 time=1ms TTL=126
Reply from 200.200.200.1: bytes=32 time<1ms TTL=126
Reply from 200.200.200.1: bytes=32 time<1ms TTL=126
Ping statistics for 200.200.200.1:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms
```

(A)

```
C:\Documents and Settings\aluno>ping -n 10 h1.diorio.corp.br
Pinging h1.diorio.corp.br [200.200.200.1] with 32 bytes of data:
Reply from 200.200.200.1: bytes=32 time=10ms TTL=126
Reply from 200.200.200.1: bytes=32 time=1ms TTL=126
Request timed out.
Reply from 200.200.200.1: bytes=32 time=4ms TTL=126
Reply from 200.200.200.1: bytes=32 time=1ms TTL=126
Reply from 200.200.200.1: bytes=32 time=1ms TTL=126
Reply from 200.200.200.1: bytes=32 time=1ms TTL=126
Reply from 200.200.200.1: bytes=32 time=1ms TTL=126
Reply from 200.200.200.1: bytes=32 time=1ms TTL=126
Reply from 200.200.200.1: bytes=32 time=1ms TTL=126
Ping statistics for 200.200.200.1:
    Packets: Sent = 10, Received = 9, Lost = 1 (10% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 10ms, Average = 2ms
```

(B)

**Figura 49.** Mensagens ICMP (via “ping”) entre o cliente multimídia H3 e o servidor multimídia H1 sem a inserção de *delays* de transmissão (A) e com a inserção de *delays* de transmissão de 200ms (B) na interface “Internet” do roteador R1.

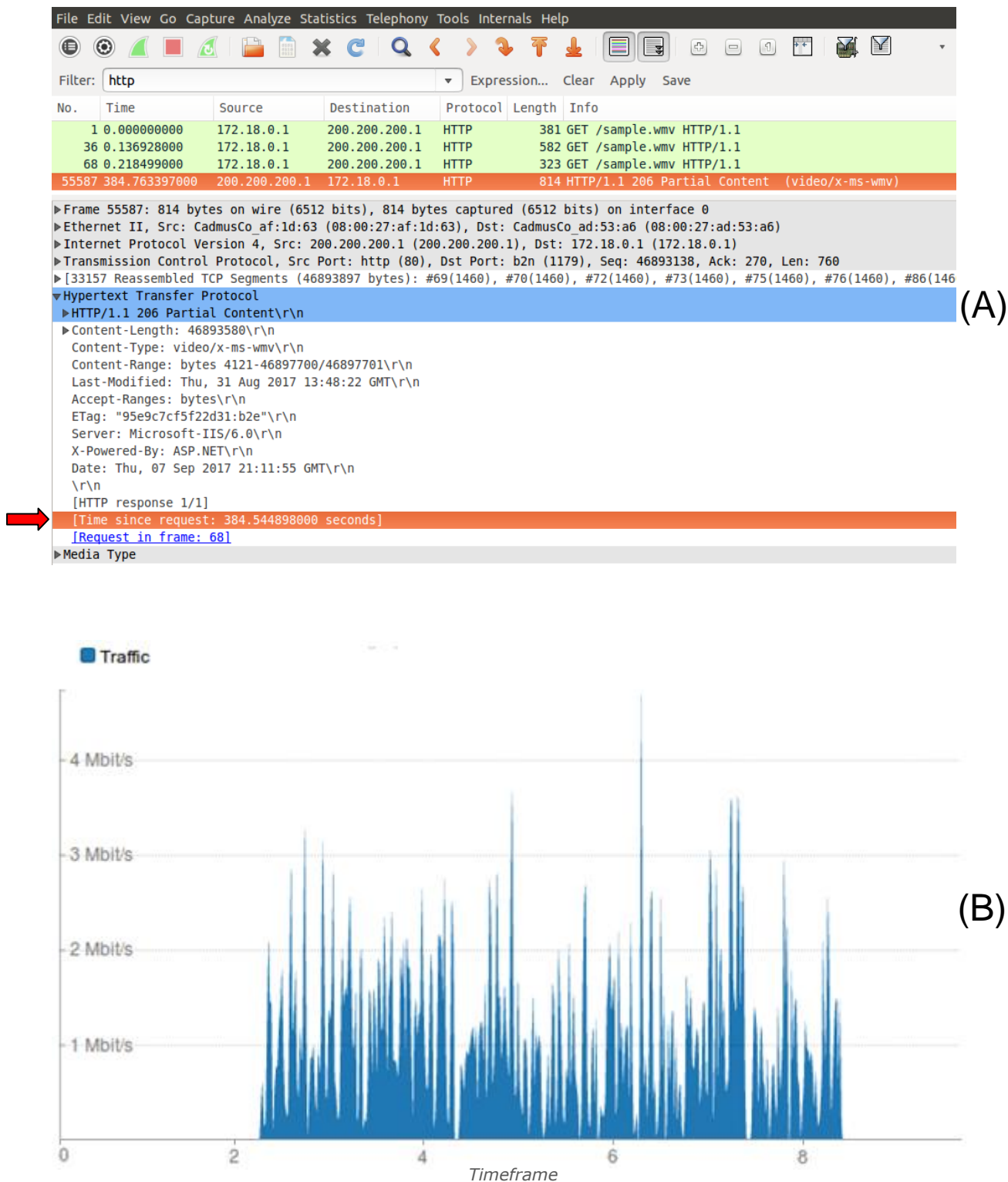
```
C:\Documents and Settings\aluno>ping h1.diorio.corp.br
Pinging h1.diorio.corp.br [200.200.200.1] with 32 bytes of data:
Reply from 200.200.200.1: bytes=32 time=4ms TTL=126
Reply from 200.200.200.1: bytes=32 time=2ms TTL=126
Reply from 200.200.200.1: bytes=32 time<1ms TTL=126
Reply from 200.200.200.1: bytes=32 time=1ms TTL=126
Ping statistics for 200.200.200.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 4ms, Average = 1ms
```

(A)

```
C:\Documents and Settings\aluno>ping h1.diorio.corp.br
Pinging h1.diorio.corp.br [200.200.200.1] with 32 bytes of data:
Reply from 200.200.200.1: bytes=32 time=206ms TTL=126
Reply from 200.200.200.1: bytes=32 time=202ms TTL=126
Reply from 200.200.200.1: bytes=32 time=201ms TTL=126
Reply from 200.200.200.1: bytes=32 time=201ms TTL=126
Ping statistics for 200.200.200.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 201ms, Maximum = 206ms, Average = 202ms
```

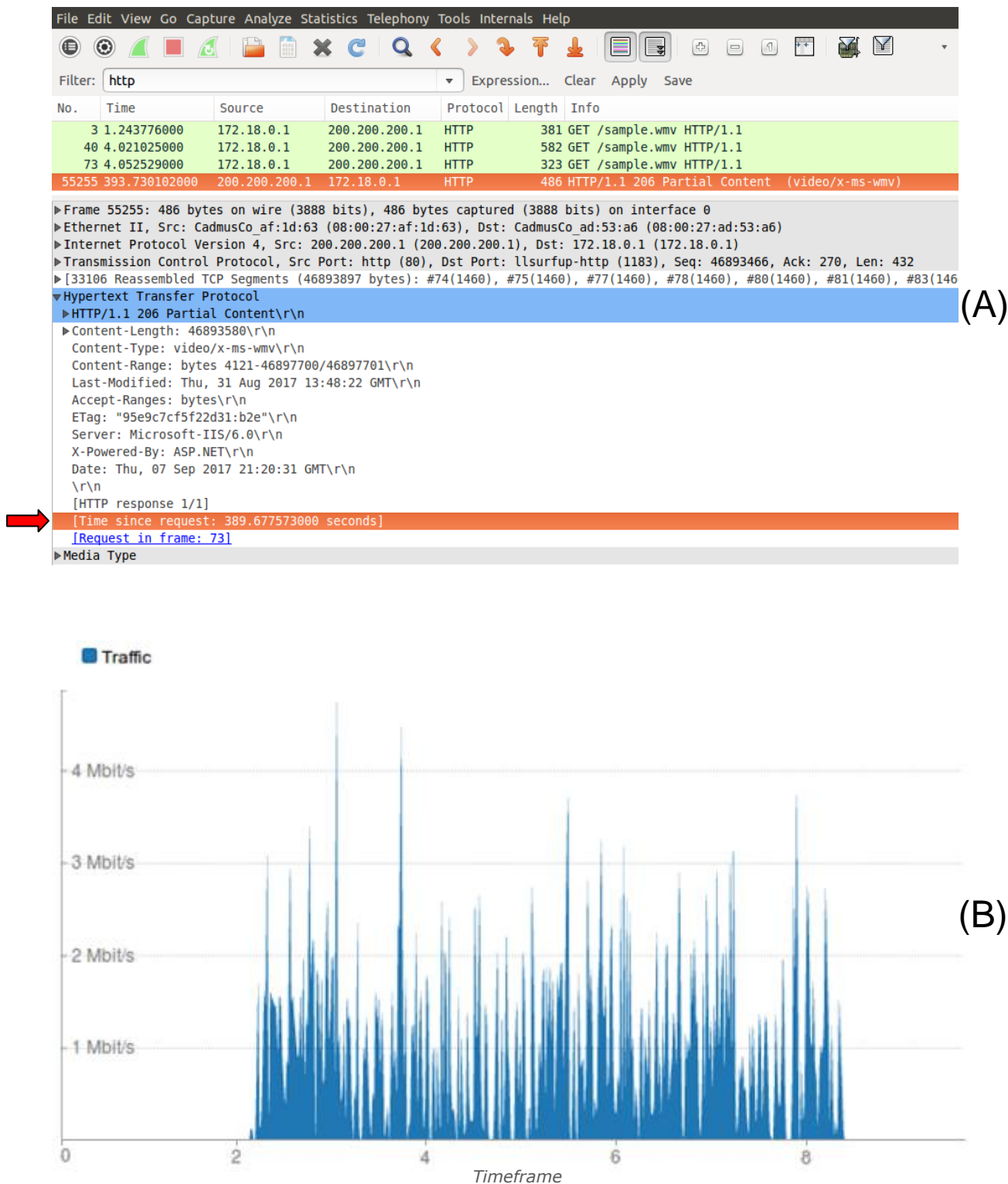
(B)

**Figura 50.** Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no primeiro acesso ao conteúdo multimídia de vídeo, sobre HTTP, sem o emprego de recursos de cache na rede local, no cenário com perdas de pacotes na ordem de 10%.

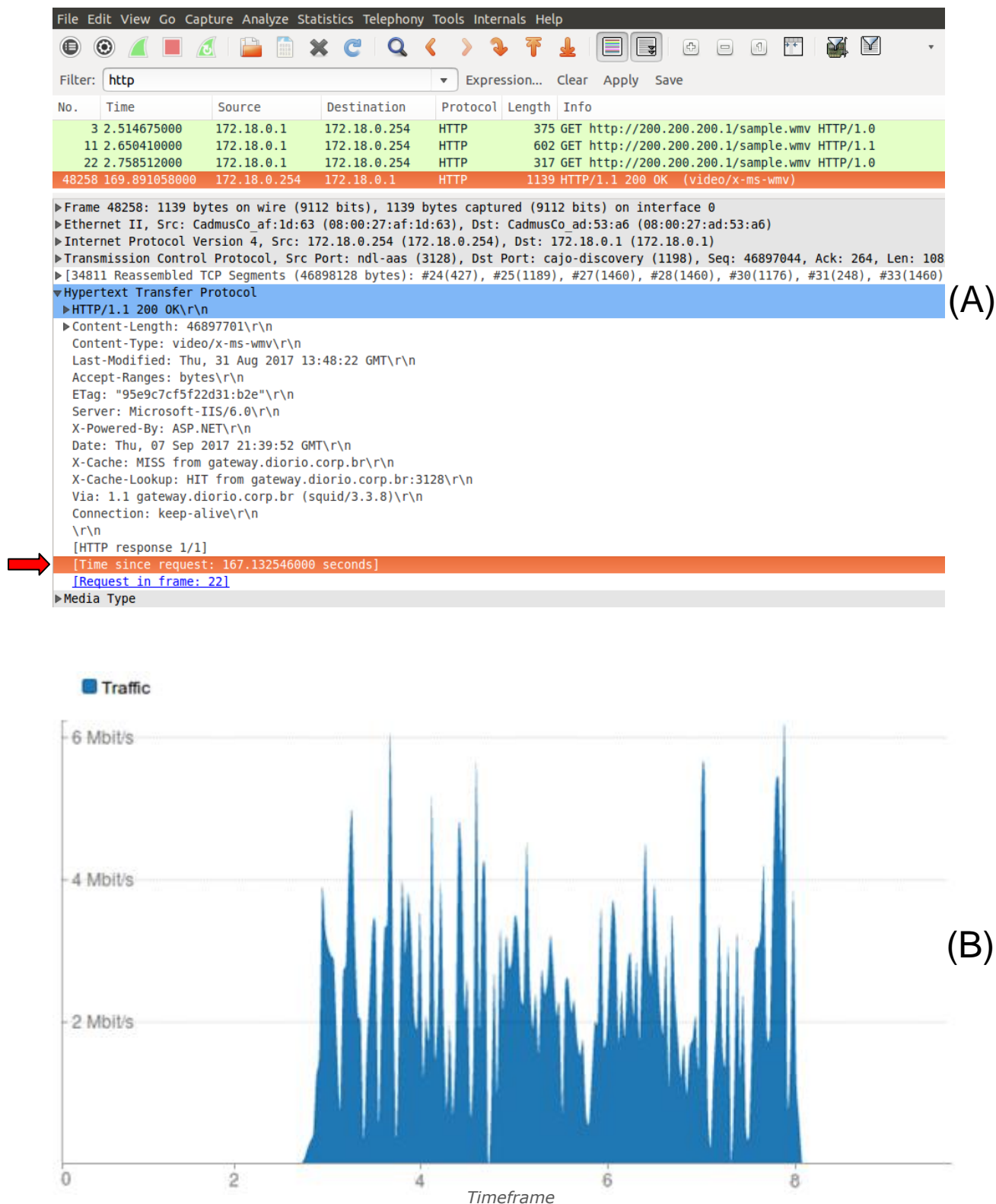




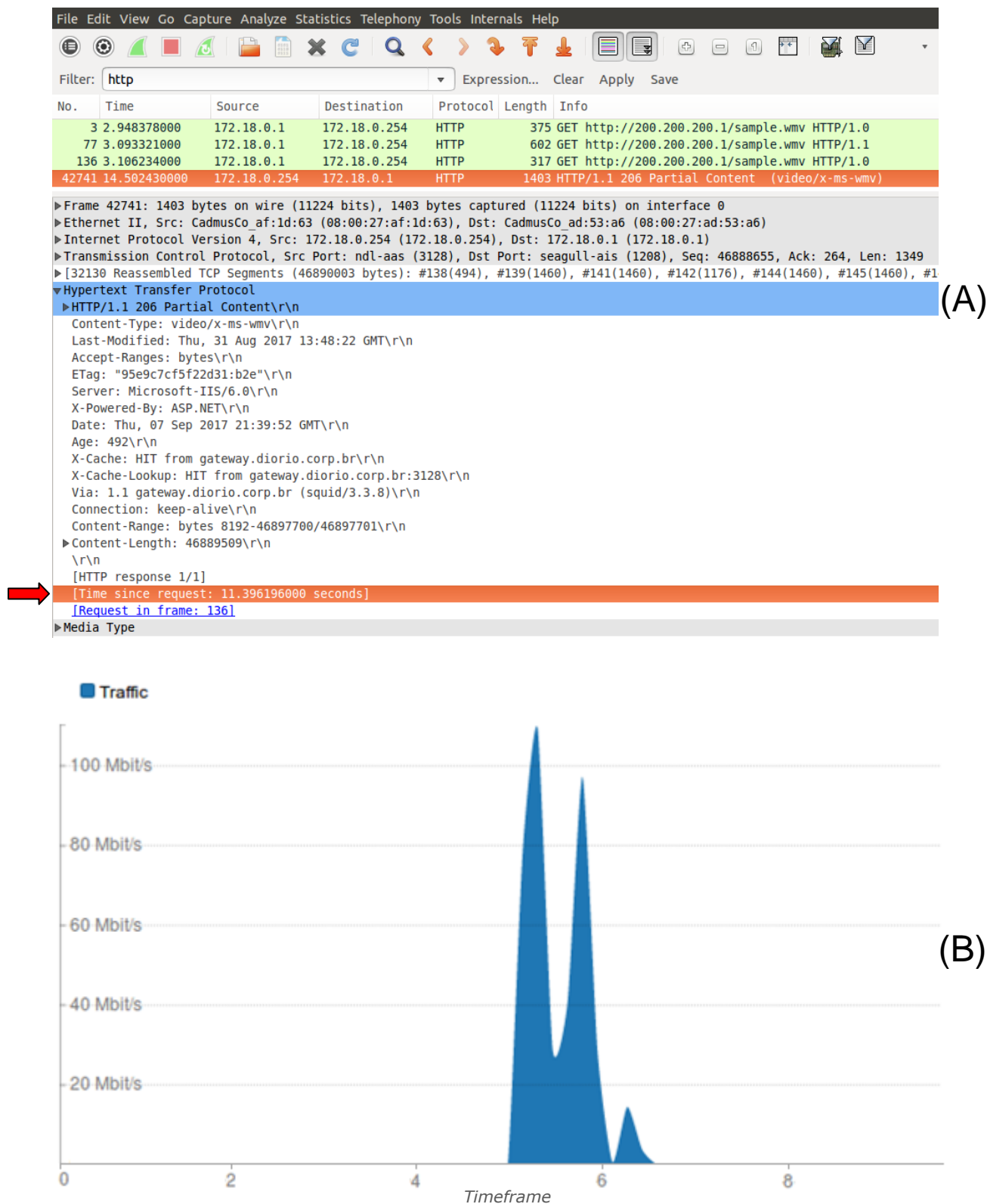
**Figura 51.** Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no segundo acesso ao conteúdo multimídia de vídeo, sobre HTTP, sem o emprego de recursos de cache na rede local, no cenário com perdas de pacotes na ordem de 10%.



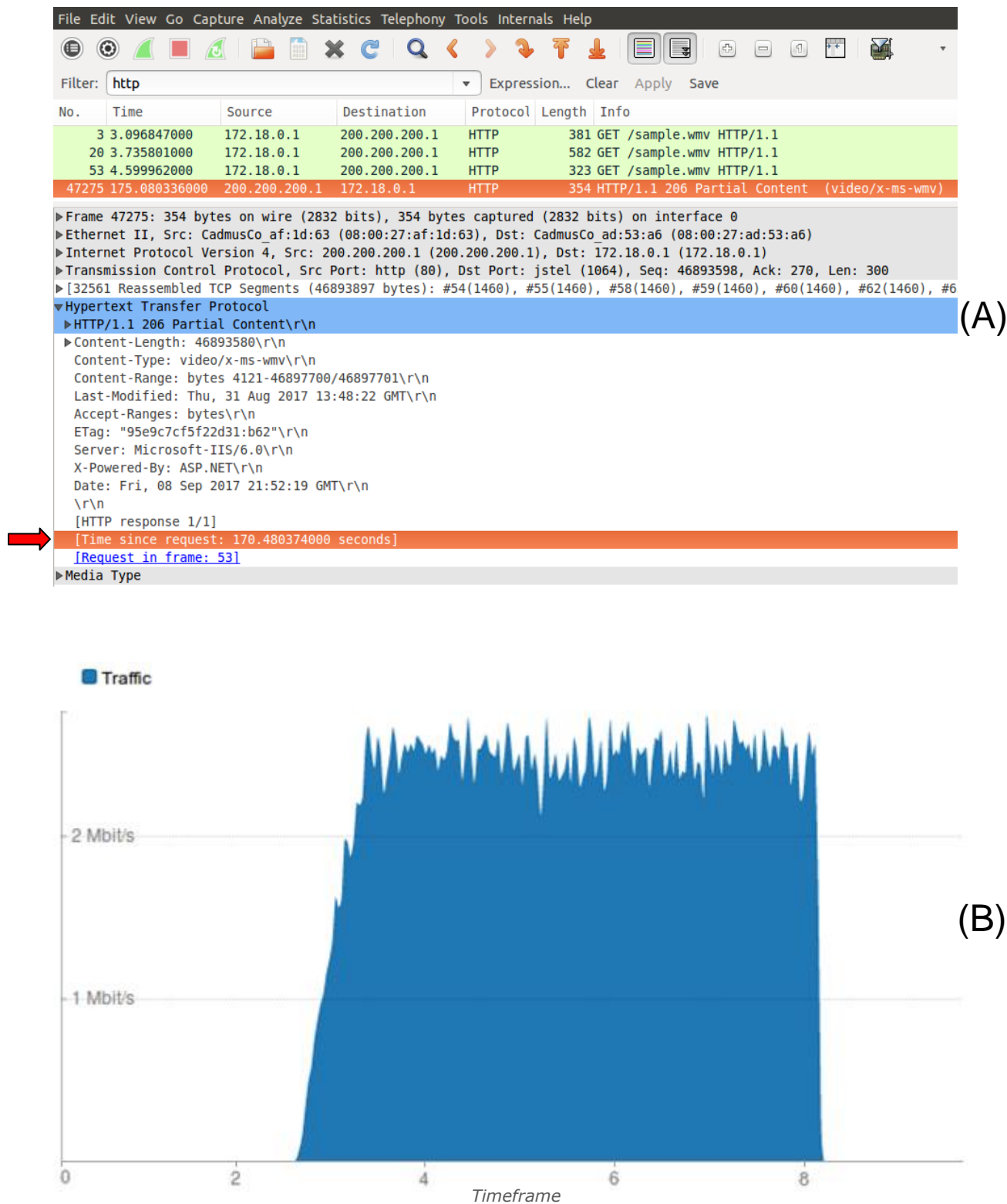
**Figura 52.** Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no primeiro acesso ao conteúdo multimídia de vídeo, sobre HTTP, com o emprego de recursos de cache junto ao *gateway* multimídia na rede local, no cenário com perdas de pacotes na ordem de 10%.



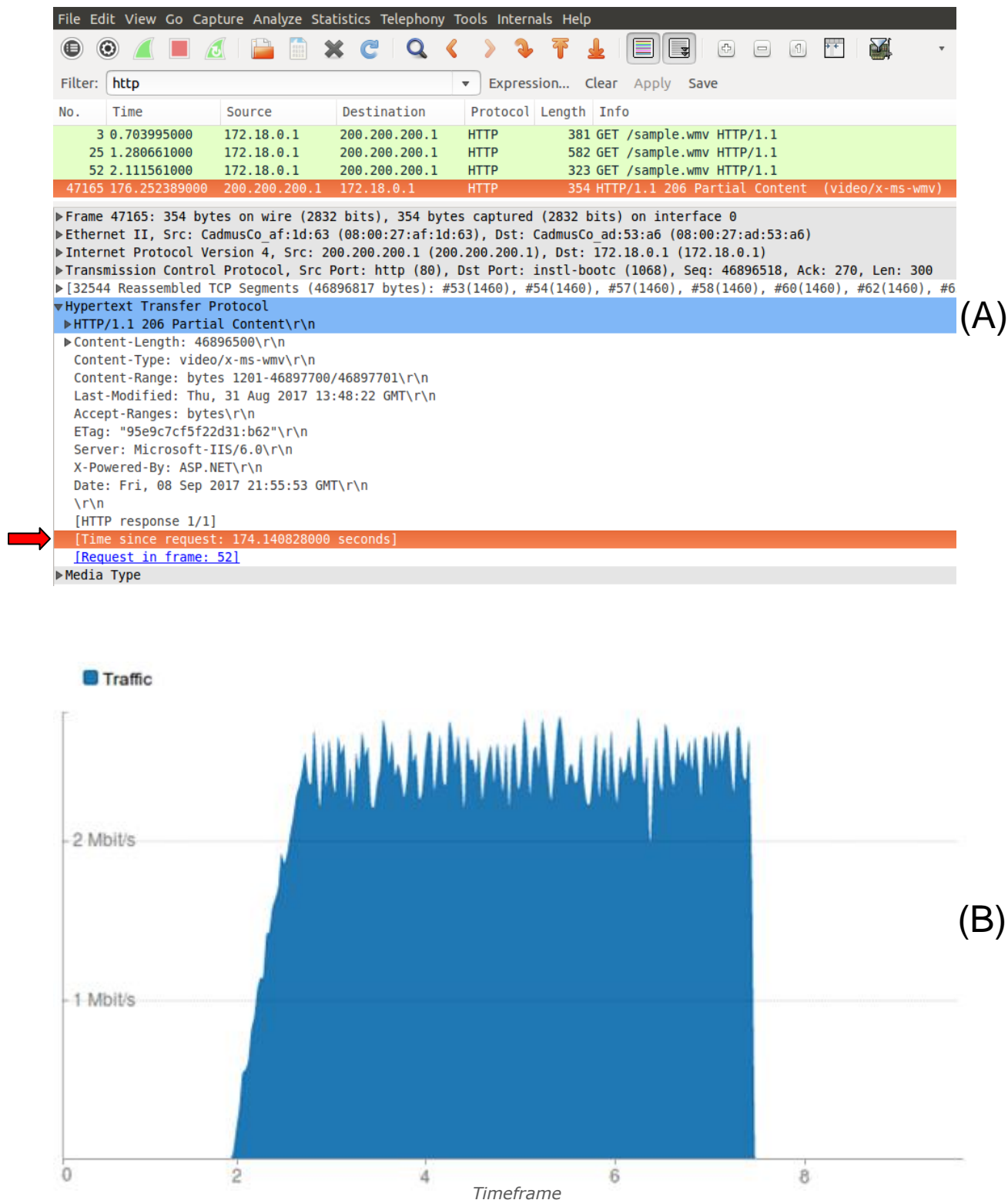
**Figura 53.** Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no segundo acesso ao conteúdo multimídia de vídeo, sobre HTTP, com o emprego de recursos de cache junto ao *gateway* multimídia na rede local, no cenário com perdas de pacotes na ordem de 10%.



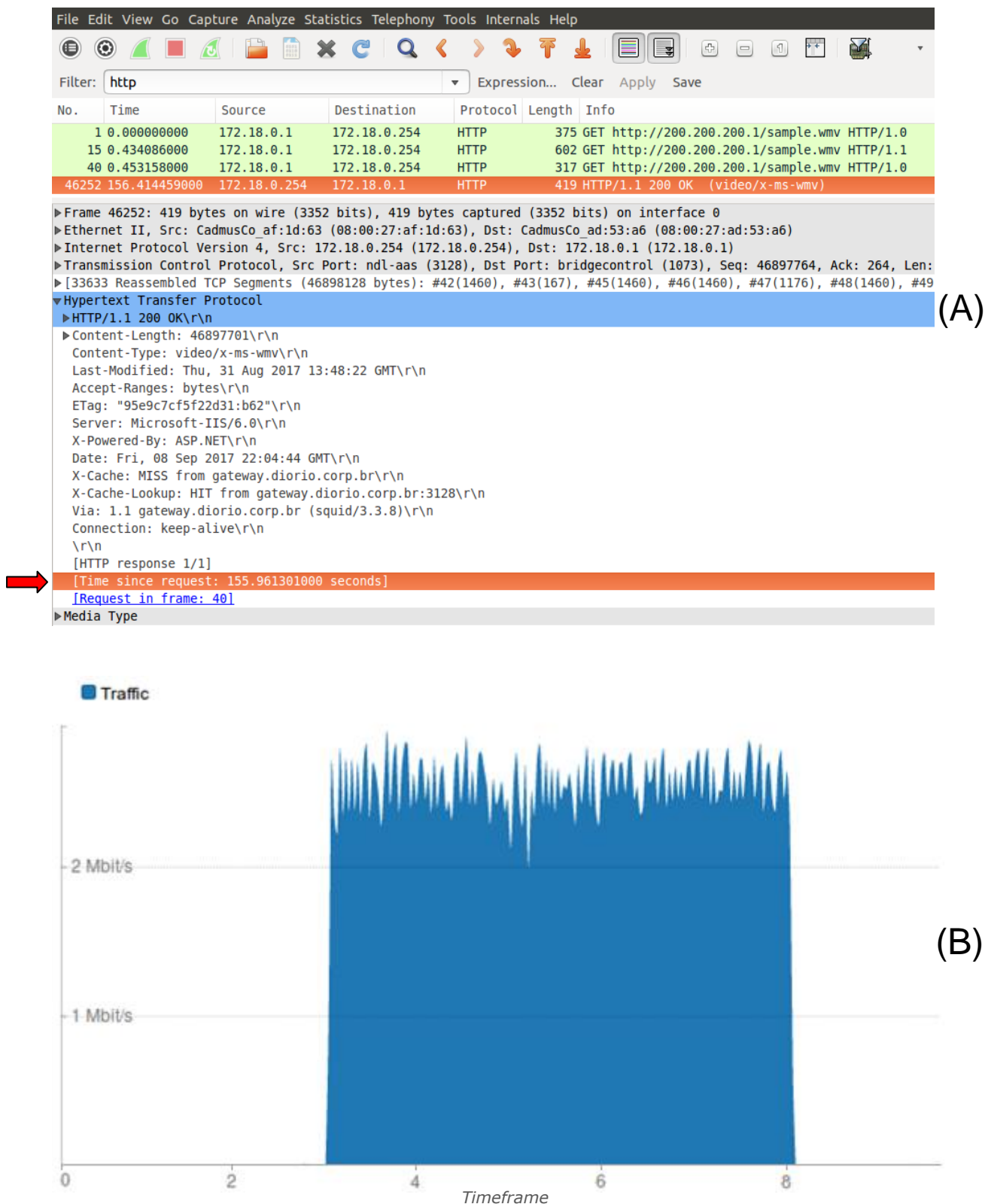
**Figura 54.** Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no primeiro acesso ao conteúdo multimídia de vídeo, sobre HTTP, sem o emprego de recursos de cache na rede local, no cenário com *delays* de transmissão na ordem de 200ms.



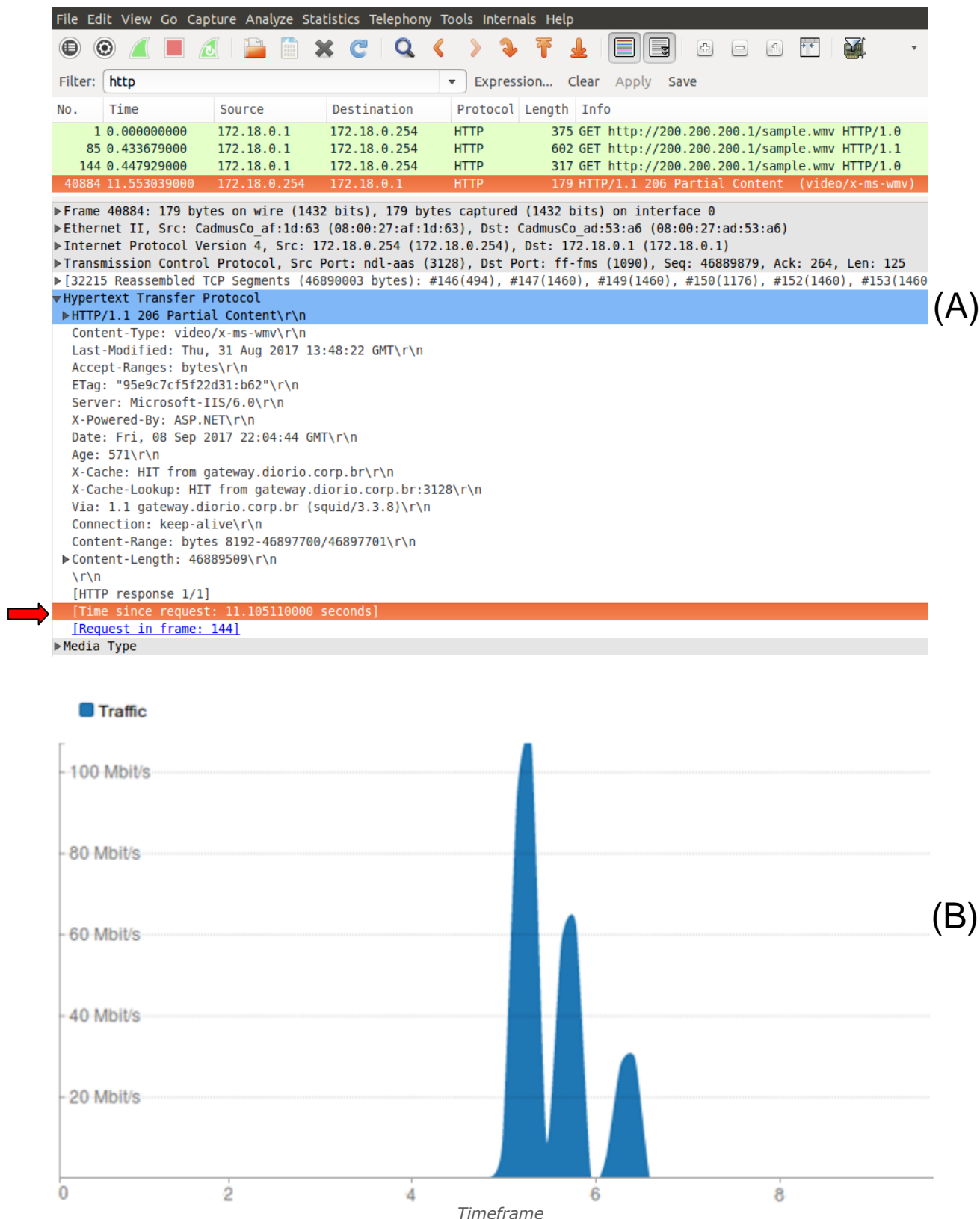
**Figura 55.** Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no segundo acesso ao conteúdo multimídia de vídeo, sobre HTTP, sem o emprego de recursos de cache na rede local, no cenário com *delays* de transmissão na ordem de 200ms.



**Figura 56.** Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no primeiro acesso ao conteúdo multimídia de vídeo, sobre HTTP, com o emprego de recursos de cache junto ao *gateway* multimídia na rede local, no cenário com *delays* de transmissão na ordem de 200ms.



**Figura 57.** Exemplo de tempo de transmissão (A) e taxa de transmissão (B) no segundo acesso ao conteúdo multimídia de vídeo, sobre HTTP, com o emprego de recursos de cache junto ao *gateway* multimídia na rede local, no cenário com *delays* de transmissão na ordem de 200ms.

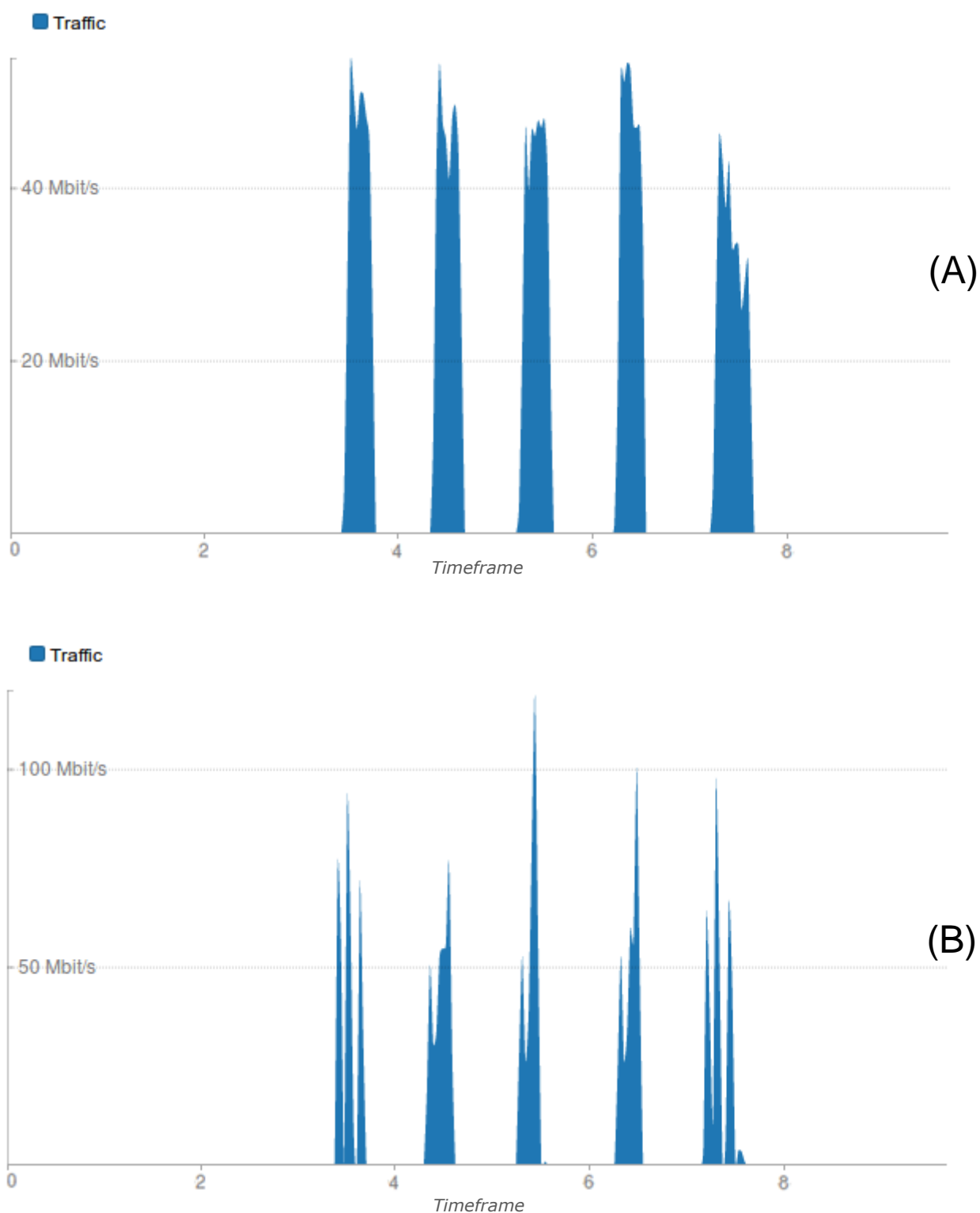


Diante de tais ilustrações, é possível observar que, em ambos os cenários (cenário com inserção de perdas de pacotes e cenário com inserção de *delays* de transmissão), os tempos de transmissão pertinentes ao conteúdo multimídia de vídeo foram melhores (inferiores) quando o recurso de cache fornecido pelo *gateway* multimídia foi empregado na rede local (especialmente, na segunda requisição ao mesmo conteúdo multimídia de vídeo em questão, isto é, com sua distribuição por meio do cache). Da mesma forma, é possível observar que as taxas de transmissão empregadas em ambas as transmissões também foram melhores (superiores) com o emprego dos recursos de cache junto ao *gateway* multimídia na rede local. Em linhas gerais, isso se deve ao fato do cliente multimídia (isto é, *player* de vídeo do cliente multimídia) estimar uma maior largura de banda entre ele e o *gateway* multimídia do que entre ele e o servidor multimídia na Internet emulada, possibilitando requisições por maiores e melhores porções de vídeo (“*video chunks*”) quando os recursos de cache estão presentes na rede local. Apenas para efeito comparativo, essa mesma observação também foi realizada por Georgopoulos et al. (2014) em seu trabalho quanto ao OpenCache em ambientes SDN, na qual o *playback client* também estimava uma maior largura de banda entre ele e a solução proposta pelos autores, também possibilitando requisições por maiores e melhores porções de vídeo em seus cenários de experimentação, de modo geral.

Nesse contexto, no âmbito experimental, essas observações foram melhores identificadas (isto é, com melhor visualização gráfica) sem as restrições de banda impostas junto ao roteador “Internet” R1, possibilitando que as requisições ao conteúdo de vídeo ultrapassassem as limitações de banda na ordem de 10 Mbps impostas no ambiente de emulação. Como exemplo, a Figura 58 ilustra as taxas de transmissão observadas em 5 requisições ao mesmo conteúdo de vídeo no ambiente de emulação, tal como realizada nas experimentações anteriores, porém sem as limitações de banda impostas junto ao roteador R1. Nesse caso, as taxas pertinentes às requisições sem o emprego de recursos de cache são ilustradas no painel superior (A) e, por sua vez, as taxas pertinentes ao emprego de recursos de cache são ilustradas no painel inferior (B). Em tal experimentação, é importante destacar que a estrutura de cache foi limpa e recriada após cada requisição ao conteúdo de vídeo, de modo que cada requisição fosse tida como uma “nova requisição inicial” (forçando a recuperação do conteúdo de vídeo junto ao servidor multimídia na “Internet” em todas as requisições). Por meio de tais ilustrações, observe que as taxas de transmissão sem o emprego de recursos de cache foram de pouco mais de 40 Mbps (próximas de 50 Mbps), enquanto que com o emprego de recursos de cache as taxas ficaram próximas de 100 Mbps.



**Figura 58.** Taxas de transmissão observadas em 5 requisições ao mesmo conteúdo multimídia de vídeo, sem o emprego de recursos de cache (A) e com o emprego de recursos de cache (B), ambas sem restrições de banda junto ao roteador R1 no ambiente de emulação. Em tal experimentação, a estrutura de cache foi limpa e recriada após cada requisição ao conteúdo de vídeo, de modo que cada requisição fosse tida como uma “nova requisição inicial”.



Nesse contexto, mesmo com a inserção de perdas de pacotes e *delays* de transmissão junto ao ambiente de emulação, as requisições iniciais por maiores e melhores porções de vídeo possibilitaram observar que até mesmo a primeira requisição empregando o recurso de cache tivesse melhores tempos e taxas de transmissão do que as requisições sem o emprego de recursos de cache na rede local. Em termos numéricos, quanto ao cenário com inserção de perdas de pacotes, os tempos de transmissão observados nas duas primeiras requisições ao mesmo conteúdo multimídia de vídeo, sem o emprego de recursos de cache, foram de aproximadamente 385 segundos e 390 segundos, respectivamente, com taxas de transmissão com picos de pouco mais de 4 Mbps em ambos os casos. Por sua vez, com o emprego de recursos de cache, os tempos de transmissão para essas mesmas requisições foram de aproximadamente 170 segundos e 11 segundos, respectivamente, com taxas de transmissão com picos de pouco mais de 6 Mbps na primeira requisição e de 100 Mbps na segunda requisição.

Por sua vez, quanto ao cenário com inserção de *delays* de transmissão, os tempos de transmissão observados nas duas primeiras requisições ao mesmo conteúdo multimídia de vídeo, sem o emprego de recursos de cache, foram de aproximadamente 170 segundos e 175 segundos, respectivamente, com taxas de transmissão na ordem aproximada de 2,5 Mbps em ambos os casos. Por outro lado, com o emprego de recursos de cache, os tempos de transmissão para essas mesmas requisições foram de aproximadamente 155 segundos e 11 segundos, respectivamente, com taxas de transmissão em torno de 2,5 Mbps na primeira requisição e de mais de 100 Mbps na segunda requisição.

Nesse contexto, de modo complementar às experimentações descritas anteriormente, é importante destacar que nenhuma configuração no âmbito da QoS foi empregada na rede local durante as experimentações pertinentes aos recursos de cache. Ainda assim, é possível observar, nos cenários com inserção de perdas de pacotes e *delays* de transmissão, que as condições da rede mediante tais situações implicaram em uma grande redução (e oscilação) nas taxas de transmissão aos clientes finais, de modo geral.

Diante de tais cenários de experimentação, é possível observar que o emprego de recursos de cache junto ao *gateway* multimídia na rede local possibilitou melhores tempos e taxas de transmissão para os conteúdos multimídia obtidos e distribuídos por meio de tal recurso para os clientes finais, com contribuições no âmbito da QoE e, também, da QoS (em especial, ao minimizar a utilização de recursos de “Internet” no segundo e demais acessos ao mesmo conteúdo multimídia em questão). Conforme descrito anteriormente, em um contexto

mais amplo, essa abordagem é de extrema importância para amenizar a utilização dos enlaces de rede entre os clientes multimídia e o ISP na qual o servidor multimídia na Internet está conectado, minimizando, por exemplo, eventuais congestionamentos, atrasos e perdas de pacotes durante esse processo de recuperação e distribuição de conteúdos multimídia, de modo geral.

#### **4.2.6 – Predição de Conteúdos Multimídia**

Conforme descrito na Seção 3.3, de modo complementar aos recursos voltados ao encaminhamento por fluxos com suporte à QoS e de cache de conteúdos multimídia, esta Tese também considera o emprego de um método de predição, baseado no filtro de Kalman e no subfiltro de Kalman, no âmbito da distribuição de conteúdos multimídia na rede. Para tal, identificadores de serviços multimídia, tais como por meio dos identificadores DSCP experimentados no ambiente de emulação, são utilizados para viabilizar (e, também, avaliar) a abordagem proposta, nesse caso, em um ambiente de simulação fornecido pelo *software* Wolfram Mathematica (versão 11).

Nesse contexto, nos mesmos moldes dos fluxos de tráfego multimídia experimentados no ambiente de emulação, duas experimentações foram realizadas no ambiente de simulação:

- (I) Simulação de transmissão de quatro serviços multimídia, tais como quanto aos serviços multimídia de dados, de áudio, de voz e de vídeo experimentados no ambiente de emulação, com um canal de comunicação empregado para cada serviço multimídia em questão;
- (II) Simulação de transmissão de quatro serviços multimídia, com quatro canais de comunicação empregados para um dos serviços multimídia (como quanto aos fluxos de dados experimentados no ambiente de emulação), três canais de comunicação empregados para dois dos serviços multimídia (como quanto aos fluxos de áudio e de vídeo experimentados no ambiente de emulação) e dois canais de comunicação para um dos serviços multimídia (como quanto aos fluxos de voz experimentados no ambiente de emulação).

Nesse cenário, para cada experimentação, as observações e predições foram realizadas a partir da geração de números aleatórios, em um espaço numérico entre 0 e 12 e por meio da função *RandomReal* do Mathematica, representando os identificadores de serviços multimídia pertinentes aos fluxos de tráfego dos serviços de dados, de áudio, de voz e de vídeo, tais como experimentados no ambiente de emulação e, nesse caso, simulados no ambiente computacional. Para tal, objetivando manter a coerência entre os ambientes de simulação e emulação empregados nesta Tese, um mapeamento entre os números aleatórios gerados no ambiente de simulação e os DSCPs empregados no ambiente de emulação, ambos quanto aos fluxos multimídia simulados/emulados em tais ambientes, foi realizado de acordo com a Tabela 12.

**Tabela 12.** Mapeamento de números aleatórios gerados no ambiente de simulação, nesse caso, representando os identificadores de serviços multimídia pertinentes aos fluxos de tráfego dos serviços de dados, de áudio, de voz e de vídeo e seus respectivos canais de comunicação, de acordo com os DSCPs empregados no ambiente de emulação.

Fluxos de tráfego multimídia simulados	Números aleatórios simulados correspondentes	Identificadores DSCP emulados correspondentes
Dados (canal 1)	$\geq 0$ e $< 1$	1 (0x01)
Áudio (canal 1)	$\geq 1$ e $< 2$	11 (0x11)
Voz (canal 1)	$\geq 2$ e $< 3$	21 (0x21)
Vídeo (canal 1)	$\geq 3$ e $< 4$	31 (0x31)
Dados (canal 2)	$\geq 4$ e $< 5$	2 (0x02)
Áudio (canal 2)	$\geq 5$ e $< 6$	12 (0x12)
Voz (canal 2)	$\geq 6$ e $< 7$	22 (0x22)
Vídeo (canal 2)	$\geq 7$ e $< 8$	32 (0x32)
Dados (canal 3)	$\geq 8$ e $< 9$	3 (0x03)
Áudio (canal 3)	$\geq 9$ e $< 10$	13 (0x13)
Vídeo (canal 3)	$\geq 10$ e $< 11$	33 (0x33)
Dados (canal 4)	$\geq 11$ e $\leq 12$	4 (0x04)

Diante desse cenário, as equações pertinentes ao filtro de Kalman e ao subfiltro de Kalman, tal como descritas na Seção 2.2 e ilustradas na Figura 3, foram implementadas no ambiente de simulação, viabilizando o método de predição baseado em identificadores de serviços multimídia descrito anteriormente. Por sua vez, no ambiente de simulação, os valores iniciais dados aos parâmetros pertinentes ao filtro de Kalman e ao subfiltro de Kalman foram

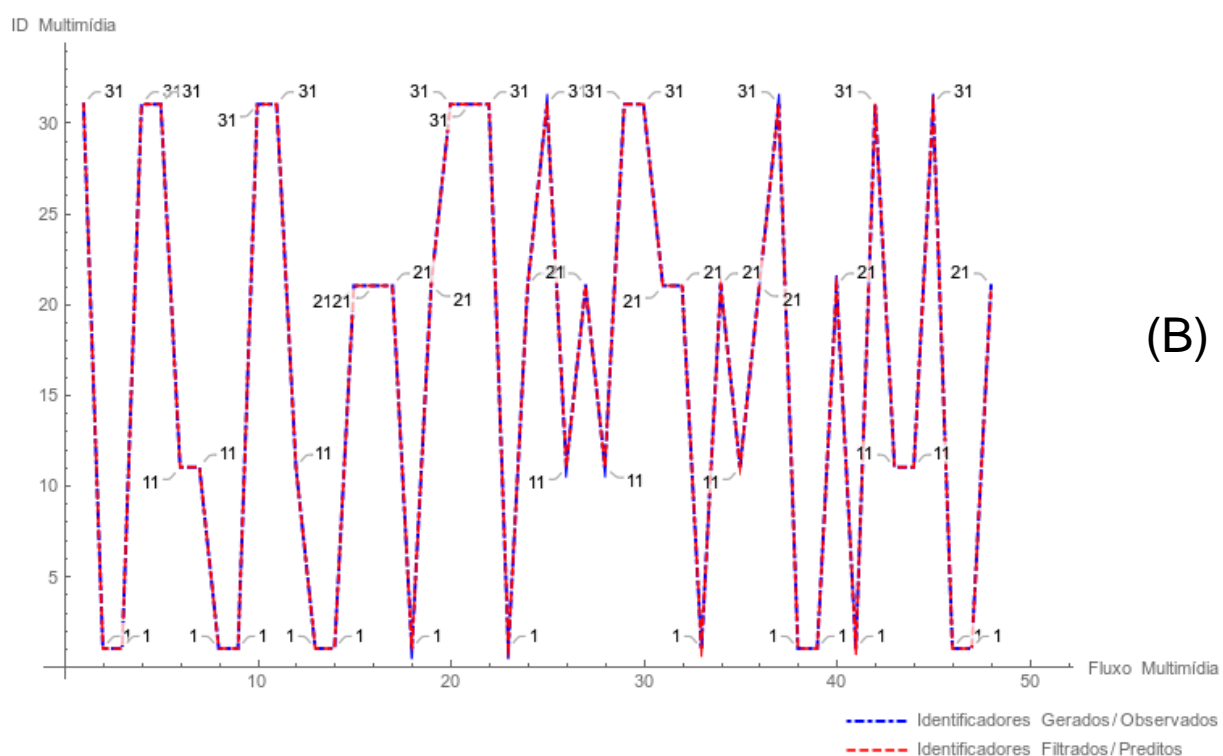
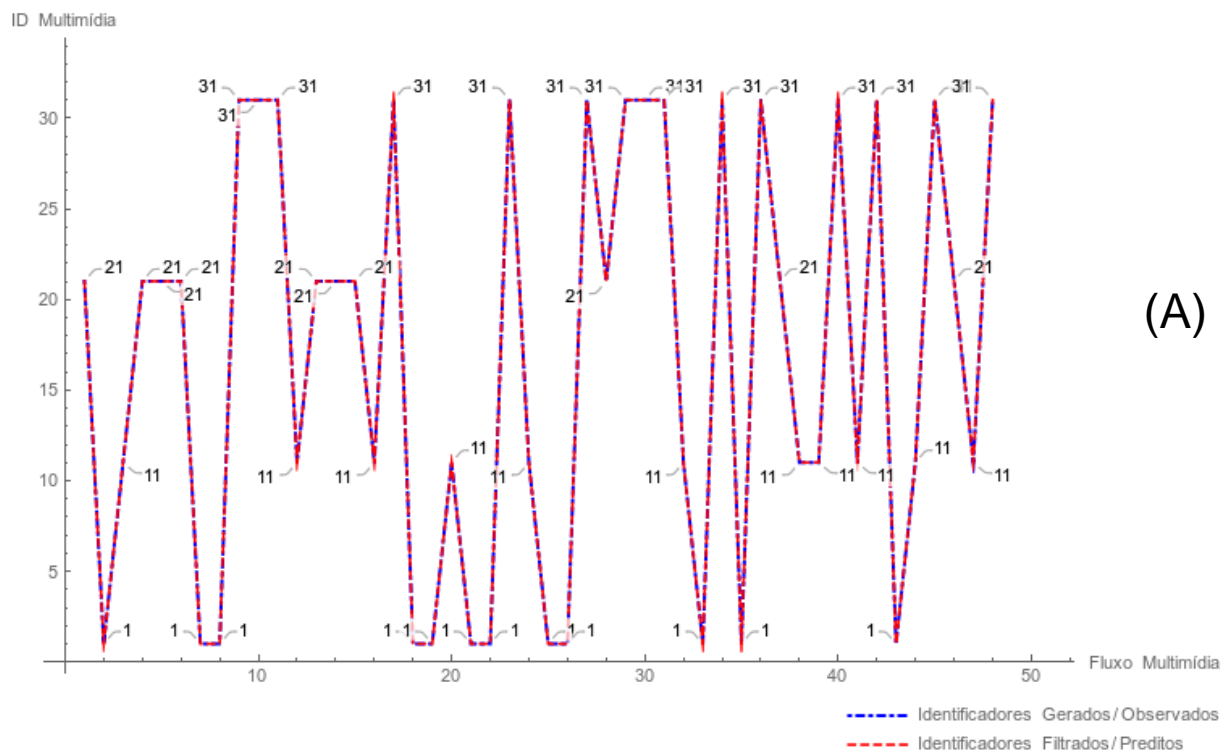
definidos conforme descrito nas Seções 2.2.1 e 2.2.2, nesse caso, com  $Y_k$  tido como *RandomReal*[12.] junto ao *software* Mathematica e com espaço numérico organizado conforme apresentado na Tabela 12. Dessa forma, todo processo de observação e filtragem/predição foi realizado considerando esse espaço numérico pertinente aos valores aleatórios gerados no ambiente computacional, posteriormente mapeado nos identificadores DSCPs pertinente a cada fluxo multimídia em questão, tal como também apresentado na Tabela 12.

Nesse cenário, é importante destacar que a simulação não considerou eventuais cargas de dados (isto é, conteúdos multimídia emulados), bem como de particularidades no âmbito da rede e das aplicações quanto ao processo de comunicação multimídia, tais como quanto ao estabelecimento de conexões, ocorrências de atrasos, perdas de pacotes e retransmissões, dentre outros, enfatizando apenas no fornecimento dos identificadores de serviços multimídia para o módulo preditor embasado no filtro de Kalman e em seu subfiltro.

Diante desse cenário, a Figura 59 ilustra os identificadores de serviços multimídia gerados/observados e os identificadores de serviços multimídia filtrados/preditos, ambos por meio do filtro de Kalman e do subfiltro de Kalman, quanto aos fluxos de tráfego simulados para os serviços multimídia de dados, de áudio, de voz e de vídeo, com um canal de comunicação empregado para cada serviço multimídia em questão, ambos junto ao ambiente de simulação. De modo complementar, a Tabela 13 apresenta os 30 primeiros números aleatórios gerados/observados no âmbito de tais identificadores de serviços multimídia, bem como seus respectivos mapeamentos quanto aos identificadores DSCPs ilustrados no painel superior (A) da Figura 59.

Por sua vez, a Figura 60 ilustra os identificadores de serviços multimídia gerados/observados e os identificadores de serviços multimídia filtrados/preditos, ambos também por meio do filtro de Kalman e do subfiltro de Kalman, quanto aos fluxos de tráfego simulados para os serviços multimídia de dados, de áudio, de voz e de vídeo, nesse caso, com quatro canais de comunicação empregados para um dos serviços multimídia, três canais de comunicação empregados para dois dos serviços multimídia e dois canais de comunicação para um dos serviços multimídia, ambos junto ao ambiente de simulação. De modo complementar, a Tabela 14 apresenta os 30 primeiros números aleatórios gerados/observados no âmbito de tais identificadores de serviços multimídia, bem como seus respectivos mapeamentos quanto aos identificadores DSCPs ilustrados no painel superior (A) da Figura 60.

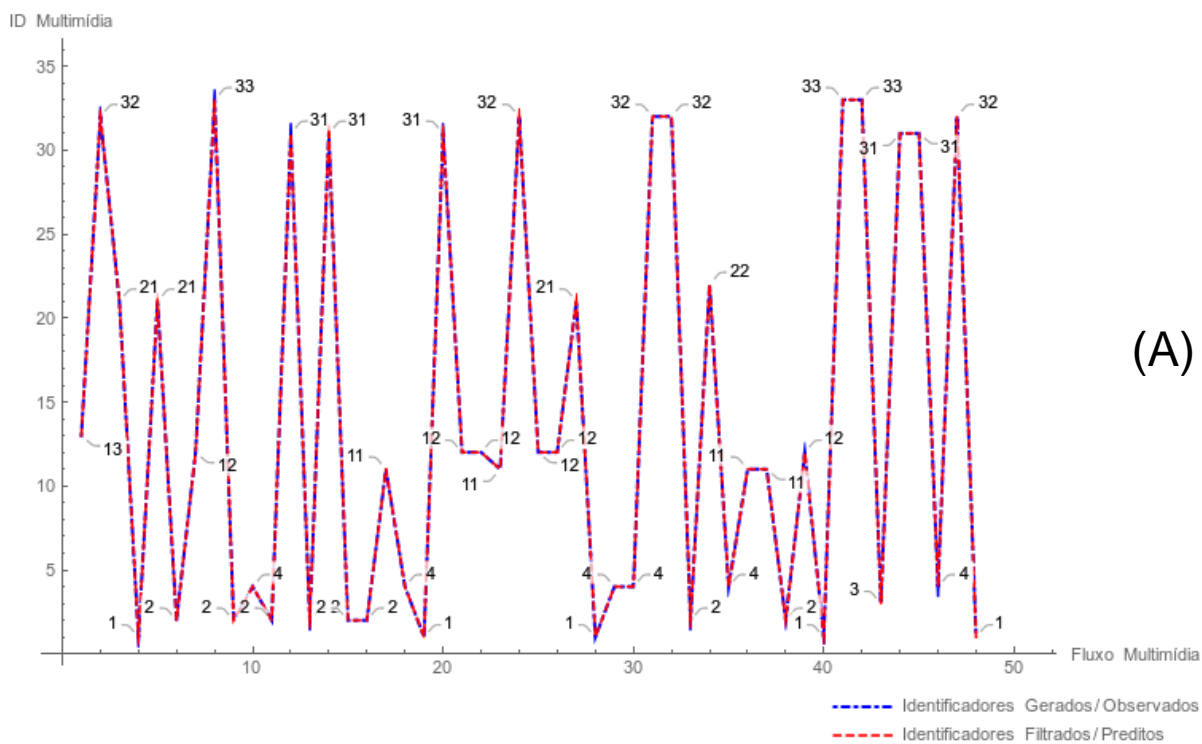
**Figura 59.** Identificadores de serviços multimídia observados e preditos no ambiente de simulação, com predições realizadas a partir de observações em dois testes distintos (A e B), ambos considerando 48 fluxos simulados aleatoriamente para quatro serviços multimídia e com um canal de comunicação para cada serviço multimídia em questão.



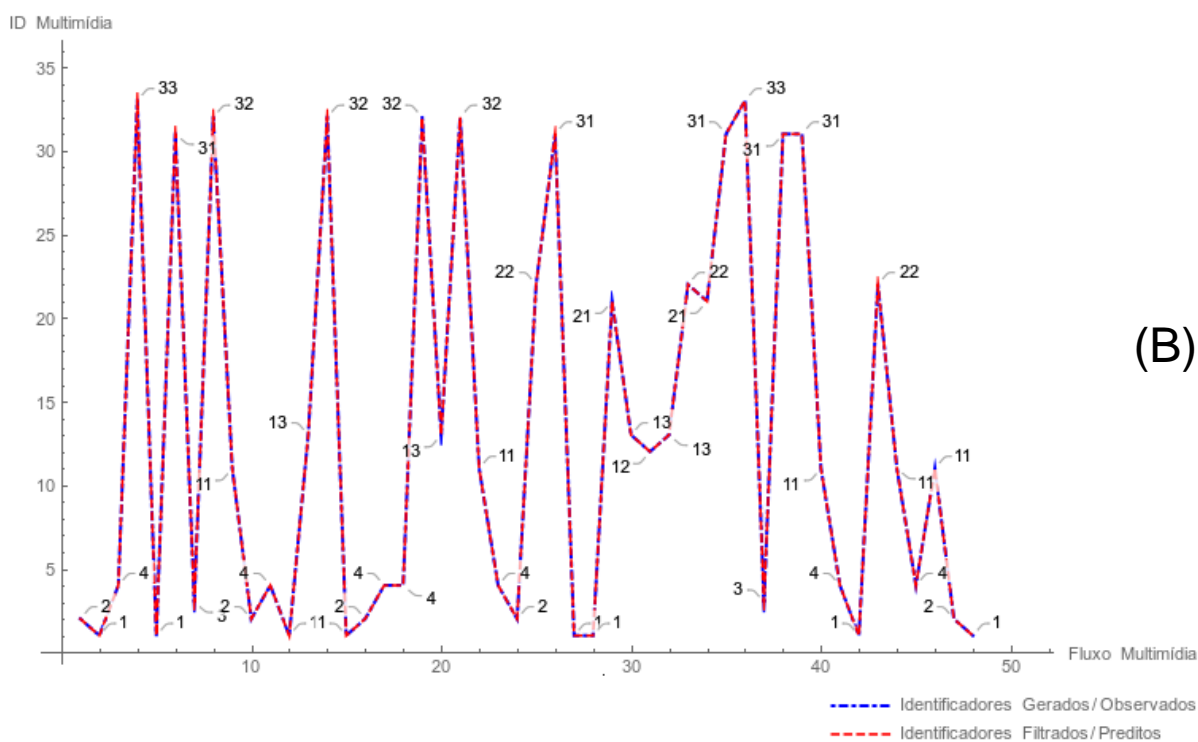
**Tabela 13.** Exemplo dos 30 primeiros números aleatórios gerados/observados no âmbito dos identificadores de serviços multimídia simulados no ambiente computacional e seus respectivos mapeamentos quanto aos identificadores DSCPs ilustrados no painel superior (A) da Figura 59.

<b>Sequência</b>	<b>Números aleatórios gerados/observados</b>	<b>Identificadores DSCPs correspondentes/mapeados</b>	<b>Fluxo de tráfego multimídia correspondente</b>
1	2.58779	21 (0x21)	Voz (canal 1)
2	0.910908	1 (0x01)	Dados (canal 1)
3	1.79899	11 (0x11)	Áudio (canal 1)
4	2.70573	21 (0x21)	Voz (canal 1)
5	2.60783	21 (0x21)	Voz (canal 1)
6	2.46614	21 (0x21)	Voz (canal 1)
7	0.664556	1 (0x01)	Dados (canal 1)
8	0.307353	1 (0x01)	Dados (canal 1)
9	3.24241	31 (0x31)	Vídeo (canal 1)
10	3.00365	31 (0x31)	Vídeo (canal 1)
11	3.19075	31 (0x31)	Vídeo (canal 1)
12	1.73374	11 (0x11)	Áudio (canal 1)
13	2.9224	21 (0x21)	Voz (canal 1)
14	2.9488	21 (0x21)	Voz (canal 1)
15	2.65029	21 (0x21)	Voz (canal 1)
16	1.76256	11 (0x11)	Áudio (canal 1)
17	3.11053	31 (0x31)	Vídeo (canal 1)
18	0.257613	1 (0x01)	Dados (canal 1)
19	0.45662	1 (0x01)	Dados (canal 1)
20	1.76278	11 (0x11)	Áudio (canal 1)
21	0.0679675	1 (0x01)	Dados (canal 1)
22	0.767265	1 (0x01)	Dados (canal 1)
23	3.00317	31 (0x31)	Vídeo (canal 1)
24	1.64292	11 (0x11)	Áudio (canal 1)
25	0.923453	1 (0x01)	Dados (canal 1)
26	0.037664	1 (0x01)	Dados (canal 1)
27	3.32141	31 (0x31)	Vídeo (canal 1)
28	2.70476	21 (0x21)	Voz (canal 1)
29	3.32034	31 (0x31)	Vídeo (canal 1)
30	3.07833	31 (0x31)	Vídeo (canal 1)

**Figura 60.** Identificadores de serviços multimídia observados e preditos no ambiente de simulação, com predições realizadas a partir de observações em dois testes distintos (A e B), ambos considerando 48 fluxos simulados aleatoriamente para quatro serviços multimídia, com quatro canais de comunicação para um dos serviços multimídia, três canais de comunicação para dois dos serviços multimídia e dois canais de comunicação para um dos serviços multimídia.



(A)



(B)



**Tabela 14.** Exemplo dos 30 primeiros números aleatórios gerados/observados no âmbito dos identificadores de serviços multimídia simulados no ambiente computacional e seus respectivos mapeamentos quanto aos identificadores DSCPs ilustrados no painel superior (A) da Figura 60.

<b>Sequência</b>	<b>Números aleatórios gerados/observados</b>	<b>Identificadores DSCPs correspondentes/mapeados</b>	<b>Fluxo de tráfego multimídia correspondente</b>
1	9.00953	13 (0x13)	Áudio (canal 3)
2	7.56949	32 (0x32)	Vídeo (canal 2)
3	2.2124	21 (0x21)	Voz (canal 1)
4	0.815193	1 (0x01)	Dados (canal 1)
5	2.86593	21 (0x21)	Voz (canal 1)
6	4.61566	2 (0x02)	Dados (canal 2)
7	5.3384	12 (0x12)	Áudio (canal 2)
8	10.2842	33 (0x33)	Vídeo (canal 3)
9	4.60619	2 (0x02)	Dados (canal 2)
10	11.68	4 (0x04)	Dados (canal 4)
11	4.06686	2 (0x02)	Dados (canal 2)
12	3.15562	31 (0x31)	Vídeo (canal 1)
13	4.67519	2 (0x02)	Dados (canal 2)
14	3.43443	31 (0x31)	Vídeo (canal 1)
15	4.97094	2 (0x02)	Dados (canal 2)
16	4.03994	2 (0x02)	Dados (canal 2)
17	1.5767	11 (0x11)	Áudio (canal 1)
18	11.5209	4 (0x04)	Dados (canal 4)
19	0.870458	1 (0x01)	Dados (canal 1)
20	3.78037	31 (0x31)	Vídeo (canal 1)
21	5.75002	12 (0x12)	Áudio (canal 2)
22	5.84695	12 (0x12)	Áudio (canal 2)
23	1.8295	11 (0x11)	Áudio (canal 1)
24	7.36313	32 (0x32)	Vídeo (canal 2)
25	5.0767	12 (0x12)	Áudio (canal 2)
26	5.74774	12 (0x12)	Áudio (canal 2)
27	2.21091	21 (0x21)	Voz (canal 1)
28	0.731052	1 (0x01)	Dados (canal 1)
29	11.8803	4 (0x04)	Dados (canal 4)
30	11.6366	4 (0x04)	Dados (canal 4)

Nesse contexto, diante dos resultados obtidos, o método de predição proposto neste trabalho mostrou-se funcional durante o processo de avaliação computacional. Os fluxos multimídia filtrados/preditos foram consistentes com os fluxos multimídia gerados/observados no ambiente de computação, nesse caso, com ambos os fluxos sendo representados por meio de seus respectivos identificadores de serviços multimídia de dados, de áudio, de voz e de vídeo. Em linhas gerais, essa abordagem pode complementar outras abordagens no âmbito da distribuição de conteúdos multimídia na rede, tais como em mecanismos de roteamento adaptativo fim-a-fim para fluxos de tráfego multimídia, em mecanismos de alocação dinâmica de banda e de QoS, bem como em mecanismos de controle preditivo de congestionamento e de análises de fluxos de tráfego, dentre outros.

De modo complementar, em um contexto mais amplo, os identificadores de serviços multimídia utilizados na experimentação e avaliação de tal método de predição (identificadores no nível de rede, via rótulos DSCP) poderiam ser substituídos por outros identificadores únicos capazes de identificar os conteúdos multimídia que serão observados e preditos, tais como por meio de identificadores no nível de aplicação (via *tags* de conteúdos na aplicação, por exemplo), dentre outros. Dessa forma, o método de predição proposto também poderia ser empregado em cenários distintos aos empregados nesta Tese, tais como para indexação, recuperação e armazenamento de conteúdos multimídia, dentre outros.

### **4.3 – Síntese e Encerramento do Capítulo**

Este Capítulo abordou os experimentos e resultados de avaliação pertinentes ao *gateway* multimídia no âmbito SDN OpenFlow. Para tal, apresentou uma visão geral sobre os ambientes e ferramentas de experimentação e avaliação empregadas nesta Tese, nesse caso, no âmbito da emulação e da simulação, bem como discorreu sobre os resultados obtidos por meio de tais ambientes, com ênfase quanto à rotulação de fluxos de tráfego multimídia, à identificação e encaminhamento diferenciado de fluxos de tráfego multimídia, às características de tráfego, tempos de transmissão, variações do atraso e perda de pacotes, ao encaminhamento por fluxos com suporte à QoS, ao cache de conteúdos multimídia e à predição de conteúdos multimídia. Em linhas gerais, os resultados obtidos mostraram que as abordagens empregadas nesta Tese, pertinentes ao *gateway* multimídia no âmbito SDN OpenFlow, são capazes de fornecer recursos para aprimorar comunicações multimídia, nesse caso, com ênfase no encaminhamento, na distribuição e na predição de conteúdos multimídia.

## 5 – Conclusão e Trabalhos Futuros

Diante dos diversos desafios e aspectos que envolvem comunicações multimídia, este trabalho discorreu sobre um *gateway* com recursos multimídia, tido como *gateway* multimídia, enquanto plataforma para aprimorar comunicações multimídia no âmbito de Redes Definidas por *Software* OpenFlow. Para tal, no âmbito de um *gateway* multimídia em SDN, os seguintes aspectos foram especialmente discutidos neste trabalho: (I) identificação, classificação e encaminhamento diferenciado de fluxos de tráfego multimídia; (II) encaminhamento por fluxos com suporte à QoS; (III) cache de conteúdos multimídia; e, (IV) predição de conteúdos multimídia.

Nesse contexto, no que se refere à identificação, classificação e encaminhamento diferenciado de fluxos de tráfego multimídia, este trabalho considerou o emprego de identificadores multimídia específicos, em especial, via rótulos DSCP, para possibilitar que múltiplos fluxos de tráfego multimídia, tais como fluxos de dados, de áudio, de voz e/ou de vídeo, por exemplo, pudessem ser classificados de acordo com seus respectivos tipos de serviços multimídia. Além disso, por meio de uma abordagem considerando uma organização flexível de *bits* pertinentes ao campo DSCP do protocolo IP, possibilitou a identificação de múltiplos e distintos canais de comunicação para um ou mais serviços multimídia na rede, tais como até dezesseis canais de comunicação para os serviços multimídia de dados, de áudio, de voz e de vídeo, por exemplo. Em linhas gerais, essa abordagem, além de explorar de uma melhor maneira o campo DSCP (evitando, por exemplo, o “desperdício” de utilização de seus *bits*), possibilitou, no protocolo IPv4, a pseudo implementação do campo “*Flow Label*” presente no cabeçalho do protocolo IPv6, viabilizando a identificação de canais distintos para, também, distintos serviços multimídia por meio de tal campo (sem a necessidade de utilização de outros campos do cabeçalho do protocolo IP, inexistentes para esse propósito no IPv4). Dessa forma, mesmo para serviços multimídia idênticos, tais como serviços de dados, de áudio, de voz e de vídeo, por exemplo, porém utilizando múltiplos canais de comunicação cada, diferentes e específicas definições de encaminhamento puderam ser fornecidas para os fluxos pertinentes a cada serviço/canal em particular, em ambos os casos, por meio do *gateway* multimídia no ambiente SDN e no âmbito das possibilidades proporcionadas pelo protocolo OpenFlow, de modo geral.

Quanto ao encaminhamento por fluxos com suporte à QoS, este trabalho abordou o emprego de múltiplas classes de tráfego, com distintas configurações em termos de alocação de banda e de priorização junto ao *gateway* multimídia no ambiente SDN OpenFlow. Além disso, de modo a maximizar a utilização dos recursos de rede, essas múltiplas e distintas classes de tráfego puderam compartilhar recursos de banda entre si, de modo que o escopo inicialmente definido para os fluxos de uma determinada classe de tráfego pudesse ser ultrapassado considerando as disponibilidades de banda de outra classe de tráfego. Nesse caso, esse “empréstimo de recursos” considerou a prioridade de cada classe de tráfego em questão, de modo que classes de maior prioridade fossem beneficiadas, antecipadamente, quando comparadas às classes de menor prioridade. Dessa forma, distintas configurações no âmbito da QoS puderam ser fornecidas para diferentes fluxos de tráfego multimídia no ambiente SDN OpenFlow, tais como para fluxos multimídia de dados e de vídeo, por exemplo, e, em ambos os casos, pela classificação de tais fluxos, por meio de seus identificadores de serviços multimídia, com suas respectivas classes de tráfego junto ao *gateway* multimídia. Além disso, considerando a identificação de diferentes canais de comunicação para um ou mais fluxos de tráfego multimídia, ambos por meio de seus respectivos identificadores de serviços multimídia, fluxos associados ao mesmo serviço multimídia, porém empregando distintos canais de comunicação, puderam receber distintas configurações no âmbito da QoS. Nesse cenário, no que se refere à QoE percebida pelo usuário final, por meio de uma métrica tradicional de QoE (nesse caso, por meio do PSNR), bem como de observações sobre os conteúdos multimídia recebidos pelos clientes finais, essa abordagem mostrou-se capaz de aprimorar a percepção de QoE do usuário final, em especial, em transmissões simultâneas de múltiplos conteúdos multimídia na rede, tais como quanto aos conteúdos multimídia de dados, de áudio e de vídeo experimentados no ambiente de emulação desta Tese, por exemplo.

No que se refere ao cache de conteúdos multimídia, este trabalho abordou o emprego de recursos de Proxy/cache para auxiliar na distribuição de conteúdos multimídia no ambiente SDN OpenFlow. Para tal, a abordagem empregada considerou o *gateway* multimídia atuando como Proxy para os clientes da SDN, centralizando as requisições realizadas por ambos e fornecendo uma estrutura de cache para aproximar o conteúdo multimídia previamente requisitado por tais clientes junto aos mesmos em seu ambiente de rede local. Dessa forma, durante o processo de entrega do conteúdo multimídia para os clientes da rede, uma cópia do mesmo foi armazenada na estrutura de cache do *gateway* multimídia, de modo

que nas próximas requisições ao mesmo conteúdo multimídia em questão, ele fosse recuperado e distribuído diretamente por meio do recurso de cache, nesse caso, sem utilizar recursos de Internet para tal. Nesse contexto, a partir de um ambiente SDN OpenFlow emulado, conteúdos multimídia de áudio, de dados e de vídeo puderam ser distribuídos aos clientes finais da SDN com melhores tempos e taxas de transmissão quando comparados aos tempos e taxas obtidos sem o emprego de tais recursos na rede, em ambos os casos, por meio de um cenário SDN OpenFlow emulado considerando um enlace Internet na ordem de 10 Mbps. De modo complementar, essa abordagem também mostrou-se eficiente inclusive em cenários com elevados percentuais em termos de perdas de pacotes (na ordem de 10%) e de *delays* de transmissão (na ordem de 200ms), com contribuições no âmbito da QoS e da QoE em ambas as experimentações.

Por sua vez, no que se refere à predição de conteúdos multimídia, uma abordagem baseada em identificadores de serviços/conteúdos multimídia foi empregada para possibilitar que futuros conteúdos multimídia (isto é, futuros fluxos com conteúdos multimídia, tais como pertinentes aos serviços de áudio, de voz, de vídeo e/ou de dados, por exemplo) pudessem ser preditos na rede. Para tal, este trabalho considerou o emprego do filtro de Kalman e do subfiltro de Kalman para viabilizar a implementação de um módulo preditor de conteúdos multimídia por meio de identificadores de serviços/conteúdos multimídia. Nesse contexto, com base em um cenário de simulação, nesse caso, considerando múltiplos serviços multimídia com múltiplos canais de comunicação associados aos mesmos, a abordagem proposta mostrou-se funcional quanto aos resultados obtidos. Os fluxos multimídia filtrados/preditos foram consistentes com os fluxos multimídia gerados/observados no ambiente de computação, nesse caso, com ambos os fluxos sendo representados por meio de seus respectivos identificadores de serviços multimídia, tal como por meio de identificadores DSCP em redes IP, por exemplo. De modo complementar, em um contexto mais amplo, esses identificadores de serviços multimídia poderiam ser substituídos por outros identificadores únicos capazes de identificar os conteúdos multimídia que serão observados e preditos, tais como por meio de identificadores no nível de aplicação (via *tags* de conteúdos na aplicação), por exemplo. Dessa forma, o método de predição proposto também poderia ser empregado em cenários distintos aos empregados nesta Tese, tais como para indexação, recuperação e armazenamento de conteúdos multimídia, dentre outros.

Dessa forma, diante dos resultados obtidos, pode-se concluir que o emprego das abordagens propostas neste trabalho, no âmbito de um *gateway* multimídia em SDN

OpenFlow, possibilitou agregar recursos para aprimorar comunicações multimídia, nesse caso, com ênfase no encaminhamento, na distribuição e na predição de conteúdos multimídia.

Quanto aos trabalhos futuros no âmbito desta Tese, é possível citar: (I) o emprego e avaliação da predição de conteúdos multimídia no âmbito do encaminhamento diferenciado de fluxos de tráfego multimídia, do cache multimídia e da QoS. Nesse caso, dentre as possibilidades de pesquisa, uma avaliação sobre a antecipação de funções administrativas, tais como no que se refere à inserção e/ou remoção de regras de fluxos junto ao controlador OpenFlow, para a manutenção ou remoção dos conteúdos multimídia armazenados em cache ou para alocações dinâmicas de banda em função da predição de futuros fluxos que serão encaminhados ao longo da rede, dentre outros, poderia ser realizado de modo a contribuir com esta Tese. Além disso, a predição de conteúdos multimídia pode ser explorada em cenários distintos aos experimentados nesta Tese, tal como para indexação, recuperação e armazenamento de conteúdos multimídia, dentre outros, o que confere possibilidades de continuidade de pesquisa no âmbito da predição multimídia em outros cenários e ambientes de rede; (II) o emprego e avaliação de *gateways* com recursos multimídia em ambientes interdomínios e utilizando recursos e funções de virtualização. Nesse caso, dentre as possibilidades de pesquisa, questões envolvendo diferentes políticas e exigências de QoS, o emprego de mecanismos de roteamento dinâmico no plano de dados da SDN e a possibilidade de múltiplos controladores OpenFlow, bem como o emprego de recursos e funções de virtualização, tais como via NFV (*Network Functions Virtualization*), por exemplo, poderiam ser explorados no âmbito de comunicações multimídia em um contexto interdomínios. Além disso, o emprego de *gateways* com recursos multimídia poderia ser explorado em cenários envolvendo centenas ou milhares de nós transmissores e receptores multimídia, tais como por meio de cenários de simulação específicos para tal; (III) o emprego e avaliação de *gateways* com recursos multimídia em ambientes SDN *wireless*. Nesse caso, dentre as possibilidades de pesquisa, questões envolvendo o emprego de recursos para o encaminhamento por fluxos com suporte à QoS, de cache e/ou de predição, por exemplo, poderiam ser explorados considerando a interoperabilidade e mobilidade do cliente sem fio entre múltiplos APs (*Access Points*) ou a partir de eventuais restrições impostas no ambiente *wireless* (tais como em termos de taxas de transmissão para um ou mais *hosts* da rede ou para um ou mais tipos de fluxos de tráfego, por exemplo). Além disso, eventuais comparações em termos de taxas de transmissão, tempos de transmissão e avaliações de QoE, dentre outras, poderiam ser realizadas entre o ambiente *wireless* e um ambiente cabeado, possibilitando, por exemplo, um

estudo comparativo entre tais ambientes; e, (IV) um estudo sobre a QoE percebida pelo usuário final em função de diferentes parâmetros de encaminhamento, de QoS e de cache multimídia. Nesse caso, dentre as possibilidades de pesquisa, diferentes estratégias de encaminhamento (tais como por meio de algoritmos de roteamento específicos, por exemplo), de escalonamento (tais como por meio de escalonadores de pacotes distintos no âmbito da QoS, por exemplo) ou de cache (tais como por meio de diferentes políticas de cache, por exemplo), poderiam ser exploradas no âmbito da QoE percebida pelo usuário final. Em linhas gerais, essas diferentes estratégias poderiam ser comparadas entre si, nesse caso, com ênfase na QoE percebida pelo usuário final.

## REFERÊNCIAS

- ADAMI, D. et al. A network control application enabling Software-Defined Quality of Service. In: IEEE International Conference on Communications (ICC), p. 6074-6079, 2015a.
- ADAMI, D. et al. Towards an SDN network control application for differentiated traffic routing. In: IEEE International Conference on Communications (ICC), p. 5827-5832, 2015b.
- ADHIKARI, V. K. et al. Measurement Study of Netflix, Hulu, and a Tale of Three CDNs. IEEE/ACM Transactions on Networking (TON), v. 23, n. 9, p. 1984-1997, 2015.
- ALMEIDA, W. et al. Caracterização da Transmissão de um Grande Evento Esportivo. In: XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), p. 627-640, 2015.
- AMELYANOVICH, A. et al. Centralized control of traffic flows in wireless LANs based on the SDN concept. In: Systems of Signal Synchronization, Generating and Processing in Telecommunications (SINKHROINFO), p. 1-5, 2017.
- BARBOSA, J. L. N.; MOURA, R. S.; SANTOS, R. L. S. Predicting Portuguese Steam Review Helpfulness Using Artificial Neural Networks. In: Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web (WebMedia), p. 287-293, 2016.
- BRITO, S. H. B. et al. An Analysis of the Largest National Ecosystem of Public Internet eXchange Points: The Case of Brazil. Journal of Communication and Information Systems, v. 31, n. 1, p. 256-271, 2016.
- BROADBENT, M. et al. OpenCache: A software-defined content caching platform. In: 1st IEEE Conference on Network Softwarization (NetSoft), p. 1-5, 2015.
- CAI, W. et al. A Survey on Cloud Gaming: Future of Computer Games. IEEE Access, v. 4, p. 7605-7620, 2016.
- CENEDESE, A.; TRAMARIN, F.; VITTURI, S. An Energy Efficient Ethernet Strategy Based on Traffic Prediction and Shaping. IEEE Transactions on Communications, v. 65, n. 1, p. 270-282, 2017.
- CHE, X.; IP, B.; LIN, L. A Survey of Current YouTube Video Characteristics. IEEE MultiMedia, v. 22, n. 2, p. 56-63, 2015.
- CHIANG, W. K.; LI, T. Y. An Extended SDN-Based In-Network Caching Service for Video on Demand. In: International Computer Symposium (ICS), p. 159-164, 2016.



CISCO SYSTEMS. Cisco Visual Networking Index: Forecast and Methodology, 2017–2021. White Paper. June 7, 2017. Disponível em: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>, 2017a.

CISCO SYSTEMS. The Zettabyte Era: Trends and Analysis. White Paper. June 7, 2017. Disponível em: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html>, 2017b.

CIUBOTARU, B.; MUNTEAN, C. H.; MUNTEAN, G. M. Mobile Multi-Source High Quality Multimedia Delivery Scheme. *IEEE Transactions on Broadcasting*, v. 63, n. 2, p. 391-403, 2017.

COSTA FILHO, R. I. T. et al. A Rede tem a Resposta: Um Modelo Escalável para Predição Integrada de Qualidade de Vídeo e QoE em Redes IP. In: XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2016.

DIAS, A. S.; WIVES, L. K.; ROESLER, V. Enhancing the Accuracy of Ratings Predictions of Video Recommender System by Segments of Interest. In: Proceedings of the 19th Brazilian Symposium on Multimedia and the Web (WebMedia), p. 241-248, 2013.

DUANMU, Z. et al. A Quality-of-Experience Index for Streaming Video. *IEEE Journal of Selected Topics in Signal Processing*, v. 11, n. 1, p. 154-166, 2017.

EGILMEZ, H. E.; TEKALP, A. M. Distributed QoS Architectures for Multimedia Streaming Over Software Defined Networks. *IEEE Transactions on Multimedia*, v. 16, n. 6, p. 1597-1609, 2014.

FOSTER, N. et al. Languages for software-defined networks. *IEEE Communications Magazine*, v. 51, n. 2, p. 128-134, 2013.

FU, S.; LIU, J.; ZHU, W. Multimedia Content Delivery with Network Function Virtualization: The Energy Perspective. *IEEE MultiMedia*, v. 24, n. 3, p. 38-47, 2017.

GEORGOPOULOS, P. et al. Using Software Defined Networking to enhance the delivery of Video-on-Demand. *Computer Communications*, v. 69, p. 79-87, 2015.

GEORGOPOULOS, P. et al. Cache as a Service: Leveraging SDN to Efficiently and Transparently Support Video-on-Demand on the Last Mile. In: 23rd International Conference on Computer Communication and Networks (ICCCN), p. 1-9, 2014.

GROSS, J. et al. Cross-layer optimization of OFDM transmission systems for MPEG-4 video streaming. *Computer Communications*, v. 27, n. 11, p. 1044-1055, 2004.

GUCK, J. W. et al. Unicast QoS Routing Algorithms for SDN: A Comprehensive Survey and Performance Evaluation. *IEEE Communications Surveys & Tutorials*, v. PP, n. 99, p. 1-29, 2017.

GUEDES, D. et al. Redes Definidas por Software: uma abordagem sistêmica para e o desenvolvimento das pesquisas em Redes de Computadores. In: *Minicursos do XXX Simpósio Brasileiro de Redes de Computadores (SBRC)*, p. 160-210, 2012.

HAKIRI, A. et al. Software-Defined Networking: Challenges and Research Opportunities for Future Internet. *Elsevier Computer Networks*, v. 75, part A, p. 453-471, 2014.

HALEPLIDIS, E. et al. Network Programmability With ForCES. *IEEE Communications Surveys & Tutorials*, v. 17, n. 3, p. 1423-1440, 2015.

HAN, G. et al. Green Routing Protocols for Wireless Multimedia Sensor Networks. *IEEE Wireless Communications*, v. 23, n. 6, p. 140-146, 2016.

HASAN, M. Z.; AL-RIZZO, H.; AL-TURJMAN, F. A Survey on Multipath Routing Protocols for QoS Assurances in Real-Time Wireless Multimedia Sensor Networks. *IEEE Communications Surveys & Tutorials*, v. 19, n. 3, p. 1424-1456, 2017.

HERBAUT, N. et al. Dynamic Deployment and Optimization of Virtual Content Delivery Networks. *IEEE MultiMedia*, v. 24, n. 3, p. 28-37, 2017.

HU, F.; HAO, Q.; BAO, K. A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation. *IEEE Communications Surveys & Tutorials*, v. 16, n. 4, p. 2181-2206, 2014.

ISHIMORI, A. et al. Control of multiple packet schedulers for improving QoS on OpenFlow/SDN networking. In: *Second European Workshop on Software Defined Networks*, p. 81-86, 2013.

JAIN, S. et al. B4: Experience with a Globally-Deployed Software Defined WAN. *ACM SIGCOMM Computer Communication Review*, v. 43, n. 4, p. 3-14, 2013.

KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. *Journal of basic Engineering*, v. 82, n. 1, p. 35-45, 1960.

- KARAKUS, M.; DURRESI, A. Quality of Service (QoS) in Software Defined Networking (SDN): A Survey. *Journal of Network and Computer Applications*, v. 80, p. 200-218, 2017.
- KARAMAN, M. A. et al. Quality of service control and resource prioritization with Software Defined Networking. In: *1st IEEE Conference on Network Softwarization (NetSoft)*, p. 1-6, 2015.
- KREUTZ, D. et al. Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, v. 103, n. 1, p. 14-76, 2015.
- KRYFTIS, Y. et al. Resource Usage Prediction Models for Optimal Multimedia Content Provision. *IEEE Systems Journal*, v. PP, n. 99, p. 1-12, 2016a.
- KRYFTIS, Y. et al. Efficient Entertainment Services Provision over a Novel Network Architecture. *IEEE Wireless Communications*, v. 23, n. 1, p. 14-21, 2016b.
- KUCMINSKI, A. et al. QoS-based Routing over Software Defined Networks. In: *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, p. 1-6, 2017.
- KUROSE, J. F.; ROSS, K. W. *Redes de Computadores e a Internet: Uma Abordagem Top-Down*. 5ª Edição. São Paulo: Pearson Addison Wesley, 2010.
- KWON, O. C.; GO, Y.; SONG, H. An Energy-Efficient Multimedia Streaming Transport Protocol over Heterogeneous Wireless Networks. *IEEE Transactions on Vehicular Technology*, v. 65, n. 8, p. 6518-6531, 2016.
- LARA, A.; KOLASANI, A.; RAMAMURTHY, B. Network Innovation using OpenFlow: A Survey. *IEEE Communications Surveys & Tutorials*, v. 16, n. 1, p. 493-512, 2014.
- LI, S. et al. Protocol Oblivious Forwarding (POF): Software-Defined Networking with Enhanced Programmability. *IEEE Network*, v. 31, n. 2, p. 58-66, 2017.
- LI, X. et al. CaaS: Caching as a Service for 5G Networks. *IEEE Access*, v. 5, p. 5982-5993, 2017.
- LIU, Y. F.; GUO, J. M.; AN, L. Multimedia Classification using Bipolar Relation Graphs. *IEEE Transactions on Multimedia*, v. 19, n. 8, p. 1860-1869, 2017.
- LÓPEZ-RODRÍGUEZ, F.; CAMPELO, D. R. A Robust SDN Network Architecture for Service Providers. In: *IEEE Global Communications Conference (GLOBECOM)*, p. 1903-1908, 2014.

MAIA, O. B.; YEHA, H. C.; ERRICO, L. A concise review of the quality of experience assessment for video streaming. *Computer Communications*, v. 57, p. 1-12, 2015.

MCKEOWN, N. et al. Openflow: Enabling Innovation In Campus Networks. *ACM SIGCOMM Computer Communication Review*, v. 38, n. 2, p. 69-74, 2008.

MENDIOLA, A. et al. A survey on the Contributions of Software-Defined Networking to Traffic Engineering. *IEEE Communications Surveys & Tutorials*, v. 19, n. 2, p. 918-953, 2017.

MIJUMBI, R. et al. Topology-Aware Prediction of Virtual Network Function Resource Requirements. *IEEE Transactions on Network and Service Management*, v. 14, n. 1, p. 106-120, 2017.

MORALES, F. et al. Virtual network topology adaptability based on data analytics for traffic prediction. *Journal of Optical Communications and Networking*, v. 9, n. 1, p. A35-A45, 2017.

MORALES, N.; BERGSTROM, M. Did the Super Bowl Kill the Internet?. *IEEE Access*, v. 5, p. 9812-9822, 2017.

MOREIRA, M. D. D. et al. Internet do Futuro: Um Novo Horizonte. In: *Minicursos do XXVII Simpósio Brasileiro de Redes de Computadores (SBRC)*, p. 1-59, 2009.

NGUYEN, V. G. et al. SDN/NFV-Based Mobile Packet Core Network Architectures: A Survey. *IEEE Communications Surveys & Tutorials*, v. 19, n. 3, p. 1567-1602, 2017.

NICHOLS, K. et al. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. *RFC 2474*, 1998.

NUNES, B. A. A. et al. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. *IEEE Communications Surveys & Tutorials*, v. 16, n. 3, p. 1617-1634, 2014.

OLIVEIRA, T. R. B. Otimização do consumo de energia em terminais móveis 3G. *Dissertação de Mestrado. Faculdade de Tecnologia (FT), Universidade Estadual de Campinas (UNICAMP), Limeira, 2011.*

OWENS, H.; DURRESI, A. Multi-Domain Video over Software-Defined Networking (MDVSDN). In: *IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*, p. 57-63, 2017.

- PAUL, S.; PAN, J.; JAIN, R. Architectures for the future networks and the next generation Internet: A survey. *Computer Communications*, v. 34, n. 1, p. 2-42, 2011.
- PEDERSEN, H. A.; DEY, S. Enhancing Mobile Video Capacity and Quality Using Rate Adaptation, RAN Caching and Processing. *IEEE/ACM Transactions on Networking*, v. 24, n. 2, p. 996-1010, 2016.
- PINOTTI, F. L. et al. An IP-based multimedia traffic generator. In: *Proceedings of the International Workshop on Telecommunications (IWT)*, 2011.
- RAWAT, D. B.; REDDY, S. R. Software Defined Networking Architecture, Security and Energy Efficiency: A survey. *IEEE Communications Surveys & Tutorials*, v. 19, n. 1, p. 325-346, 2017.
- REZENDE, C. et al. A Reactive and Scalable Unicast Solution for Video Streaming over VANETs. *IEEE Transactions on Computers*, v. 64, n. 3, p. 614-626, 2015.
- RIOS NETO, A.; KUGA, H. Estimaco adaptativa do ruído no estado para o filtro de Kalman. In: *4th Congresso Brasileiro de Automática*, Campinas, Brasil, 1982.
- ROBITZA, W. et al. Challenges of future multimedia QoE monitoring for internet service providers. *Multimedia Tools and Applications*, p. 1-24, 2017.
- SAFADI, B.; DERBAS, N.; QUÉNOT, G. Descriptor optimization for multimedia indexing and retrieval. *Multimedia Tools and Applications*, v. 74, n. 4, p. 1267-1290, 2015.
- SAID, O. et al. IoT-RTP and IoT-RTCP: Adaptive Protocols for Multimedia Transmission over Internet of Things Environments. *IEEE Access*, v. 5, p. 16757-16773, 2017.
- SANTOS, B. et al. Caracterizaco do Comportamento dos Clientes de Vídeo ao Vivo Durante um Evento de Larga Escala. In: *XXXIV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, p. 468-481, 2016.
- SEDDIKI, M. S. et al. FlowQoS: QoS for the rest of us. In: *Proceedings of the third workshop on Hot topics in software defined networking (HotSDN '14)*, p. 207-208, 2014.
- SEDDIKI, M. S. et al. FlowQoS: Per-Flow Quality of Service for Broadband Access Networks. Georgia Institute of Technology, SCS Technical Report, GT-CS-15-02. Disponível em <https://smartech.gatech.edu/handle/1853/53190>, 2015.

- SILVA, D. P. et al. Uma Arquitetura para o Aproveitamento de QoS Interdomínios em Redes Virtuais baseadas no OpenFlow. In: Proceedings of the 31st Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), p. 893-906, 2013.
- SILVA-JUNIOR, O.; KAMIENSKI, C. A.; LORENA, A. C. Predição de Fluxos em Redes de Computadores. In: XXXII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), p. 119-132, 2014.
- SUN, L. et al. Beyond 1 Million Nodes: A Crowdsourced Video Content Delivery Network. IEEE MultiMedia, v. 24, n. 3, p. 54-63, 2017.
- TANENBAUM, A. S. Redes de Computadores. 4ª Edição. Rio de Janeiro: Campus, 2003.
- TRAJANO, A. F.; FERNANDEZ, M. P. ContentSDN: A Content-Based Transparent Proxy Architecture in Software-Defined Networking. In: IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), p. 532-539, 2016.
- URSINI, E. L. et al. n-Steps Ahead Software Reliability Prediction using the Kalman Filter. Applied Mathematics and Computation, v. 245, p. 116-134, 2014.
- VARELA, M. et al. QoE in the Web: A dance of design and performance. In: Seventh International Workshop on Quality of Multimedia Experience (QoMEX), p. 1-7, 2015.
- WELCH, G.; BISHOP, G. An Introduction to the Kalman Filter. Course 8, ACM SIGGRAPH, 2001.
- WINKLER, S.; MOHANDAS, P. The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics. IEEE Transactions on Broadcasting, v. 54, n. 3, p. 660-668, 2008.
- Xia, W. et al. A Survey on Software-Defined Networking. IEEE Communications Surveys & Tutorials, v. 17, n. 1, p. 27-51, 2015.
- YAP, K. K. et al. Taking the Edge off with Espresso: Scale, Reliability and Programmability for Global Internet Peering. In: Proceedings of the Conference of the ACM Special Interest Group on Data Communication, p. 432-445, 2017.
- ZHAI, D. et al. Supervised Distributed Hashing for Large-Scale Multimedia Retrieval. IEEE Transactions on Multimedia, v. PP, n. 99, p. 1-12, 2017.
- ZHAO, T.; LIU, Q.; CHEN, C. W. QoE in Video Transmission: A User Experience-Driven Strategy. IEEE Communications Surveys & Tutorials, v. 19, n. 1, p. 285-302, 2017.

## Apêndice A – Regras de Fluxos

### A.1 – Rotulação de Fluxos de Tráfego Multimídia

De modo complementar à Seção 4.2.1, esta Seção discorre sobre as regras de fluxos empregadas para viabilizar a rotulação de todos os fluxos de tráfego multimídia experimentados no ambiente de emulação. Nesse contexto, como exemplo, a Figura 61, que é pertinente a listagem de regras de fluxos junto ao controlador OpenFlow, ilustra as definições empregadas para viabilizar a rotulação dos fluxos de dados pertinentes aos canais 1 e 2 do gerador de tráfego multimídia.

**Figura 61.** Definições empregadas para viabilizar a rotulação de fluxos de dados pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação.

```

"data-1": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->1,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x6",
    "ipv4_dst": "172.18.0.4",
    "ipv4_src": "200.200.200.2",
    "tcp_dst": "6001"
  },
  [...]
  "version": "OF_13"
}

"data-2": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->2,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x6",
    "ipv4_dst": "172.18.0.5",
    "ipv4_src": "200.200.200.2",
    "tcp_dst": "6002"
  },
  [...]
  "version": "OF_13"
}

```

Nesse exemplo, quanto às informações ilustradas na Figura 61, os nomes “*data-1*” e “*data-2*” referem-se aos nomes utilizados para identificar as duas regras de fluxos em questão junto ao controlador OpenFlow. Nesse caso, “*data-1*” refere-se aos fluxos de dados no canal 1 e “*data-2*” refere-se aos fluxos de dados no canal 2, ambos pertinentes ao gerador de tráfego multimídia. Por sua vez, em ambas as regras de fluxos, instruções distintas de rotulação são definidas de acordo com seus respectivos *match fields* OpenFlow. Nesse caso, na regra de fluxo “*data-1*”, a instrução é rotular os fluxos de tráfego utilizando o identificador DSCP 0x01, equivalente ao decimal 1. Para tal, essa regra é aplicada aos fluxos gerados pelo servidor multimídia H2 e destinados ao cliente multimídia H6 na porta 6001/TCP, nesse caso, utilizando informações dos protocolos IPv4 e TCP. Por sua vez, na regra de fluxo “*data-2*”, a instrução é rotular os fluxos de tráfego utilizando o identificador DSCP 0x02 (decimal 2). Para tal, essa regra é aplicada aos fluxos gerados pelo servidor multimídia H2 e destinados ao cliente multimídia H7 na porta 6002/TCP. Em ambas as regras, a versão empregada do protocolo OpenFlow é a 1.3.

Em tal exemplo, é importante destacar que apenas as informações pertinentes aos nomes das regras de fluxos e suas respectivas definições em termos de instruções (ações) e *match fields*, bem como quanto a versão empregada do protocolo OpenFlow, são ilustradas em tal figura. As demais informações, tidas como não relevantes para o exemplo em questão, foram substituídas e ilustradas pela *string* “[...]”.

Nesse contexto, de modo similar à Figura 61, as Figuras 62 e 63 ilustram, respectivamente, as definições empregadas para viabilizar a rotulação dos fluxos de dados pertinentes ao serviço de hospedagem *web* (via HTTP) e de transferência de arquivos (via FTP) no ambiente de emulação. Por meio de tais ilustrações, observe que as definições pertinentes ao serviço de hospedagem *web* são descritas na regra de fluxo “*data-http-1*” e, por sua vez, as definições pertinentes ao serviço de transferência de arquivos são descritas nas regras de fluxos “*data-ftp-1*” e “*data-ftp-2*”. Em ambos os casos, as regras são aplicadas aos fluxos gerados pelo servidor multimídia H1 e destinados ao cliente multimídia H4, porém com identificadores DSCP 0x03 (decimal 3) para os fluxos de dados sobre HTTP (com origem na porta 80/TCP do servidor multimídia H1) e DSCP 0x04 (decimal 4) para os fluxos de dados sobre FTP (com origem nas portas 20/TCP e 21/TCP do servidor multimídia H1).



**Figura 62.** Definições empregadas para viabilizar a rotulação de fluxos de dados pertinentes ao serviço de hospedagem *web* sobre HTTP no ambiente de emulação.

```

"data-http-1": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->3,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x6",
    "ipv4_dst": "172.18.0.2",
    "ipv4_src": "200.200.200.1",
    "tcp_src": "80"
  },
  [...]
  "version": "OF_13"
}

```

**Figura 63.** Definições empregadas para viabilizar a rotulação de fluxos de dados pertinentes ao serviço de transferência de arquivos sobre FTP no ambiente de emulação.

```

"data-ftp-1": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->4,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x6",
    "ipv4_dst": "172.18.0.2",
    "ipv4_src": "200.200.200.1",
    "tcp_src": "20"
  },
  [...]
  "version": "OF_13"
}

"data-ftp-2": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->4,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x6",
    "ipv4_dst": "172.18.0.2",
    "ipv4_src": "200.200.200.1",
    "tcp_src": "21"
  },
  [...]
  "version": "OF_13"
}

```

Quanto aos fluxos de áudio pertinentes aos canais 1 e 2 do gerador de tráfego multimídia, a Figura 64 ilustra as definições utilizadas para viabilizar a rotulação de ambos no ambiente de emulação. Nesse caso, “*audio-1*” refere-se aos fluxos de áudio no canal 1 e “*audio-2*” refere-se aos fluxos de áudio no canal 2. Em tais exemplos, na regra de fluxo “*audio-1*”, a instrução é rotular os fluxos de tráfego utilizando o identificador DSCP 0x11 (decimal 17). Para tal, essa regra é aplicada aos fluxos gerados pelo servidor multimídia H2 e destinados ao cliente multimídia H6 na porta 7001/UDP. Por sua vez, na regra de fluxo “*audio-2*”, a instrução é rotular os fluxos de tráfego utilizando o identificador DSCP 0x12 (decimal 18). Para tal, essa regra é aplicada aos fluxos gerados pelo servidor multimídia H2 e destinados ao cliente multimídia H7 na porta 7002/UDP.

**Figura 64.** Definições empregadas para viabilizar a rotulação de fluxos de áudio pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação.

```

"audio-1": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->17,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x11",
    "ipv4_dst": "172.18.0.4",
    "ipv4_src": "200.200.200.2",
    "udp_dst": "7001"
  },
  [...]
  "version": "OF_13"
}

"audio-2": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->18,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x11",
    "ipv4_dst": "172.18.0.5",
    "ipv4_src": "200.200.200.2",
    "udp_dst": "7002"
  },
  [...]
  "version": "OF_13"
}

```

No que se refere aos fluxos de áudio pertinentes ao serviço de *streaming* de áudio, as Figuras 65 e 66 ilustram as definições utilizadas para viabilizar a rotulação de ambos no ambiente de emulação. Em tais ilustrações, a Figura 65 é pertinente às definições relativas ao protocolo RTP (com regras nomeadas como “*audio-rtp-1*” e “*audio-rtp-2*”) e, por sua vez, a Figura 66 é pertinente às definições relativas ao protocolo RTSP (com regras nomeadas como “*audio-rtsp-1*” e “*audio-rtsp-2*”).

**Figura 65.** Definições empregadas para viabilizar a rotulação de fluxos de áudio pertinentes ao serviço de *streaming* de áudio sobre RTP no ambiente de emulação.

```

    "audio-rtp-1": {
      [...]
      "instructions": {
        "instruction_apply_actions": {
          "actions": "set_field=ip_dscp->19,output=normal"
        }
      },
      "match": {
        "eth_type": "0x800",
        "ip_proto": "0x11",
        "ipv4_dst": "172.18.0.3",
        "ipv4_src": "200.200.200.1",
        "udp_src": "5004"
      },
      [...]
      "version": "OF_13"
    }
  {
    "audio-rtp-2": {
      [...]
      "instructions": {
        "instruction_apply_actions": {
          "actions": "set_field=ip_dscp->19,output=normal"
        }
      },
      "match": {
        "eth_type": "0x800",
        "ip_proto": "0x11",
        "ipv4_dst": "172.18.0.3",
        "ipv4_src": "200.200.200.1",
        "udp_src": "5005"
      },
      [...]
      "version": "OF_13"
    }
  }

```

**Figura 66.** Definições empregadas para viabilizar a rotulação de fluxos de áudio pertinentes ao serviço de *streaming* de áudio sobre RTSP no ambiente de emulação.

```

"audio-rtsp-1": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->19,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x6",
    "ipv4_dst": "172.18.0.3",
    "ipv4_src": "200.200.200.1",
    "tcp_src": "554"
  },
  [...]
  "version": "OF_13"
}

"audio-rtsp-2": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->19,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x11",
    "ipv4_dst": "172.18.0.3",
    "ipv4_src": "200.200.200.1",
    "udp_src": "554"
  },
  [...]
  "version": "OF_13"
}

```

Em tais exemplos (Figuras 65 e 66), a instrução de ambas as regras é rotular os fluxos de tráfego utilizando o identificador DSCP 0x13 (decimal 19). Para tal, as regras são aplicadas aos fluxos gerados pelo servidor multimídia H1 e destinados ao cliente multimídia H5, nesse caso, nas portas de comunicação 5004/UDP (regra de fluxo “*audio-rtp-1*”), 5005/UDP (regra de fluxo “*audio-rtp-2*”), 554/TCP (regra de fluxo “*audio-rtsp-1*”) e 554/UDP (regra de fluxo “*audio-rtsp-2*”). Assim como nas demais regras de fluxos exemplificadas anteriormente, a versão empregada do protocolo OpenFlow nas definições de ambas as regras de fluxos em questão é a 1.3.

Quanto aos fluxos de voz pertinentes aos canais 1 e 2 do gerador de tráfego multimídia, a Figura 67 ilustra as definições utilizadas para viabilizar a rotulação de ambos no ambiente de emulação. Nesse caso, “*voice-1*” refere-se aos fluxos de voz no canal 1 e “*voice-2*” refere-se aos fluxos de voz no canal 2. Em tais exemplos, na regra de fluxo “*voice-1*”, a instrução é rotular os fluxos de tráfego utilizando o identificador DSCP 0x21 (decimal 33). Para tal, essa regra é aplicada aos fluxos gerados pelo servidor multimídia H2 e destinados ao cliente multimídia H6 na porta 8001/UDP. Por sua vez, na regra de fluxo “*voice-2*”, a instrução é rotular os fluxos de tráfego utilizando o identificador DSCP 0x22 (decimal 34). Para tal, essa regra é aplicada aos fluxos gerados pelo servidor multimídia H2 e destinados ao cliente multimídia H7 na porta 8002/UDP.

**Figura 67.** Definições empregadas para viabilizar a rotulação de fluxos de voz pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação.

```

"voice-1": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->33,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x11",
    "ipv4_dst": "172.18.0.4",
    "ipv4_src": "200.200.200.2",
    "udp_dst": "8001"
  },
  [...],
  "version": "OF_13"
}

"voice-2": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->34,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x11",
    "ipv4_dst": "172.18.0.5",
    "ipv4_src": "200.200.200.2",
    "udp_dst": "8002"
  },
  [...],
  "version": "OF_13"
}

```

No que se refere aos fluxos de vídeo pertinentes aos canais 1 e 2 do gerador de tráfego multimídia, a Figura 68 ilustra as definições utilizadas para viabilizar a rotulação de ambos no ambiente de emulação. Nesse caso, “*video-1*” refere-se aos fluxos de vídeo no canal 1 e “*video-2*” refere-se aos fluxos de vídeo no canal 2. Em tais exemplos, na regra de fluxo “*video-1*”, a instrução é rotular os fluxos de tráfego utilizando o identificador DSCP 0x31 (decimal 49). Para tal, essa regra é aplicada aos fluxos gerados pelo servidor multimídia H2 e destinados ao cliente multimídia H6 na porta 9001/UDP. Por sua vez, na regra de fluxo “*video-2*”, a instrução é rotular os fluxos de tráfego utilizando o identificador DSCP 0x32 (decimal 50). Para tal, essa regra é aplicada aos fluxos gerados pelo servidor multimídia H2 e destinados ao cliente multimídia H7 na porta 9002/UDP.

**Figura 68.** Definições empregadas para viabilizar a rotulação de fluxos de vídeo pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação.

```

"video-1": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->49,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x11",
    "ipv4_dst": "172.18.0.4",
    "ipv4_src": "200.200.200.2",
    "udp_dst": "9001"
  },
  [...]
  "version": "OF_13"
}

"video-2": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->50,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x11",
    "ipv4_dst": "172.18.0.5",
    "ipv4_src": "200.200.200.2",
    "udp_dst": "9002"
  },
  [...]
  "version": "OF_13"
}

```

Por sua vez, quanto aos fluxos de vídeo pertinentes ao serviço de *streaming* de vídeo, as Figuras 69 e 70 ilustram as definições utilizadas para viabilizar a rotulação de ambos no ambiente de emulação. Em tais ilustrações, a Figura 69 é pertinente às definições relativas ao protocolo RTP (com regras nomeadas como “*video-rtp-1*” e “*video-rtp-2*”) e, por sua vez, a Figura 70 é pertinente às definições relativas ao protocolo RTSP (com regras nomeadas como “*video-rtsp-1*” e “*video-rtsp-2*”).

**Figura 69.** Definições empregadas para viabilizar a rotulação de fluxos de vídeo pertinentes ao serviço de *streaming* de vídeo sobre RTP no ambiente de emulação.

```

"video-rtp-1": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->51,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x11",
    "ipv4_dst": "172.18.0.1",
    "ipv4_src": "200.200.200.1",
    "udp_src": "5004"
  },
  [...]
  "version": "OF_13"
}

{
  "video-rtp-2": {
    [...]
    "instructions": {
      "instruction_apply_actions": {
        "actions": "set_field=ip_dscp->51,output=normal"
      }
    },
    "match": {
      "eth_type": "0x800",
      "ip_proto": "0x11",
      "ipv4_dst": "172.18.0.1",
      "ipv4_src": "200.200.200.1",
      "udp_src": "5005"
    },
    [...]
    "version": "OF_13"
  }
}

```

**Figura 70.** Definições empregadas para viabilizar a rotulação de fluxos de vídeo pertinentes ao serviço de *streaming* de vídeo sobre RTSP no ambiente de emulação.

```

"video-rtsp-1": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->51,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x6",
    "ipv4_dst": "172.18.0.1",
    "ipv4_src": "200.200.200.1",
    "tcp_src": "554"
  },
  [...]
  "version": "OF_13"
}

"video-rtsp-2": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ip_dscp->51,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_proto": "0x11",
    "ipv4_dst": "172.18.0.1",
    "ipv4_src": "200.200.200.1",
    "udp_src": "554"
  },
  [...]
  "version": "OF_13"
}

```

Em tais exemplos (Figuras 69 e 70), a instrução de ambas as regras é rotular os fluxos de tráfego utilizando o identificador DSCP 0x33 (decimal 51). Para tal, as regras são aplicadas aos fluxos gerados pelo servidor multimídia H1 e destinados ao cliente multimídia H3, nesse caso, nas portas 5004/UDP (regra “*video-rtp-1*”), 5005/UDP (regra “*video-rtp-2*”), 554/TCP (regra “*video-rtsp-1*”) e 554/UDP (regra “*video-rtsp-2*”).

Diante desse cenário, todos os fluxos de tráfego multimídia em questão foram devidamente rotulados de acordo com seus respectivos identificadores DSCP no ambiente de emulação, tal como ilustrado por meio das Figuras 71 (fluxos de dados via MTG), 72 (fluxos de dados via HTTP e FTP), 73 (fluxos de áudio via MTG), 74 (fluxos de áudio via RTP e RTSP), 75 (fluxos de voz via MTG), 76 (fluxos de vídeo via MTG) e 77 (fluxos de vídeo via RTP e RTSP). Nesse caso, MTG refere-se ao gerador de tráfego multimídia.



**Figura 71.** Fluxos de tráfego multimídia de dados nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, e seus respectivos identificadores DSCP no ambiente de emulação.

**(A)**

Filter: `tcp.port==6001` Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
24	40.224667	200.200.200.2	172.18.0.4	TCP	74	33936 > 6001 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK
25	40.224758	172.18.0.4	200.200.200.2	TCP	74	6001 > 33936 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MS
27	40.239349	200.200.200.2	172.18.0.4	TCP	66	33936 > 6001 [ACK] Seq=1 Ack=1 Win=29248 Len=0 TSval=1
28	40.246983	200.200.200.2	172.18.0.4	TCP	691	[TCP segment of a reassembled PDU]
29	40.247011	172.18.0.4	200.200.200.2	TCP	66	6001 > 33936 [ACK] Seq=1 Ack=626 Win=30336 Len=0 TSval
30	40.247873	200.200.200.2	172.18.0.4	TCP	66	33936 > 6001 [FIN, ACK] Seq=626 Ack=1 Win=29248 Len=0

Frame 27: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:85:dc:eb (08:00:27:85:dc:eb)

Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.4 (172.18.0.4)

Version: 4  
Header Length: 20 bytes

Differentiated Services Field: 0x04 (DSCP 0x01: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

0000 01.. = Differentiated Services Codepoint: Unknown (0x01)

.... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)

Total Length: 52  
Identification: 0x95bc (38332)

Flags: 0x02 (Don't Fragment)  
Fragment offset: 0  
Time to live: 62  
Protocol: TCP (6)

Header checksum: 0xa6a22 [correct]  
Source: 200.200.200.2 (200.200.200.2)  
Destination: 172.18.0.4 (172.18.0.4)

Transmission Control Protocol, Src Port: 33936 (33936), Dst Port: 6001 (6001), Seq: 1, Ack: 1, Len: 0

**(B)**

Filter: `tcp.port==6002` Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
6	27.715515	200.200.200.2	172.18.0.5	TCP	74	54516 > 6002 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK
7	27.715600	172.18.0.5	200.200.200.2	TCP	74	6002 > 54516 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MS
10	27.743650	200.200.200.2	172.18.0.5	TCP	66	54516 > 6002 [ACK] Seq=1 Ack=1 Win=29248 Len=0 TSval=1
11	27.760852	200.200.200.2	172.18.0.5	TCP	691	[TCP segment of a reassembled PDU]
12	27.760953	172.18.0.5	200.200.200.2	TCP	66	6002 > 54516 [ACK] Seq=1 Ack=626 Win=30336 Len=0 TSval
15	27.797494	200.200.200.2	172.18.0.5	TCP	66	54516 > 6002 [FIN, ACK] Seq=626 Ack=1 Win=29248 Len=0

Frame 10: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:c6:13:07 (08:00:27:c6:13:07)

Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.5 (172.18.0.5)

Version: 4  
Header Length: 20 bytes

Differentiated Services Field: 0x08 (DSCP 0x02: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

0000 10.. = Differentiated Services Codepoint: Unknown (0x02)

.... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)

Total Length: 52  
Identification: 0x09a2 (2466)

Flags: 0x02 (Don't Fragment)  
Fragment offset: 0  
Time to live: 62  
Protocol: TCP (6)

Header checksum: 0xf637 [correct]  
Source: 200.200.200.2 (200.200.200.2)  
Destination: 172.18.0.5 (172.18.0.5)

Transmission Control Protocol, Src Port: 54516 (54516), Dst Port: 6002 (6002), Seq: 1, Ack: 1, Len: 0

**Figura 72.** Fluxos de tráfego multimídia de dados sobre HTTP (A) e sobre FTP (B), pertinentes aos serviços de hospedagem *web* e de transferência de arquivos, respectivamente, e seus respectivos identificadores DSCP no ambiente de emulação.

**(A)**

No.	Time	Source	Destination	Protocol	Length	Info
4	0.063247000	172.18.0.2	200.200.200.1	HTTP	339	GET / HTTP/1.1
5	0.091950000	200.200.200.1	172.18.0.2	HTTP	763	HTTP/1.1 200 OK (text/html)
7	3.593764000	172.18.0.2	200.200.200.1	HTTP	381	GET /sample.zip HTTP/1.1
8664	15.970806000	200.200.200.1	172.18.0.2	HTTP	1007	HTTP/1.1 200 OK (application/x-zip-compressed)
8668	29.463525000	172.18.0.2	200.200.200.1	HTTP	339	GET / HTTP/1.1
8669	29.487076000	200.200.200.1	172.18.0.2	HTTP	763	HTTP/1.1 200 OK (text/html)

Filter: **http** Expression... Clear Apply

Frame 8664: 1007 bytes on wire (8056 bits), 1007 bytes captured (8056 bits)

Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:f5:fe:76 (08:00:27:f5:fe:76)

Internet Protocol Version 4, Src: 200.200.200.1 (200.200.200.1), Dst: 172.18.0.2 (172.18.0.2)

Version: 4  
Header Length: 20 bytes  
Differentiated Services Field: 0x0c (DSCP 0x03: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
**0000 11.. = Differentiated Services Codepoint: Unknown (0x03)**  
... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
Total Length: 993  
Identification: 0x4a3d (19005)  
Flags: 0x02 (Don't Fragment)  
Fragment offset: 0  
Time to live: 126  
Protocol: TCP (6)  
Header checksum: 0x71ef [correct]  
Source: 200.200.200.1 (200.200.200.1)  
Destination: 172.18.0.2 (172.18.0.2)

Transmission Control Protocol, Src Port: 80 (80), Dst Port: 1035 (1035), Seq: 8366394, Ack: 613, Len: 953

[5820 Reassembled TCP Segments (8366637 bytes): #8(1460), #9(1460), #10(1460), #12(1460), #13(1460), #15(1460), #16(1460), #17(1460), #18(1460), #19(1460), #20(1460), #21(1460), #22(1460), #23(1460), #24(1460), #25(1460), #26(1460), #27(1460), #28(1460), #29(1460), #30(1460), #31(1460), #32(1460), #33(1460), #34(1460), #35(1460), #36(1460), #37(1460), #38(1460), #39(1460), #40(1460), #41(1460), #42(1460), #43(1460), #44(1460), #45(1460), #46(1460), #47(1460), #48(1460), #49(1460), #50(1460), #51(1460), #52(1460), #53(1460), #54(1460), #55(1460), #56(1460), #57(1460), #58(1460), #59(1460), #60(1460), #61(1460), #62(1460), #63(1460), #64(1460), #65(1460), #66(1460), #67(1460), #68(1460), #69(1460), #70(1460), #71(1460), #72(1460), #73(1460), #74(1460), #75(1460), #76(1460), #77(1460), #78(1460), #79(1460), #80(1460), #81(1460), #82(1460), #83(1460), #84(1460), #85(1460), #86(1460), #87(1460), #88(1460), #89(1460), #90(1460), #91(1460), #92(1460), #93(1460), #94(1460), #95(1460), #96(1460), #97(1460), #98(1460), #99(1460), #100(1460), #101(1460), #102(1460), #103(1460), #104(1460), #105(1460), #106(1460), #107(1460), #108(1460), #109(1460), #110(1460), #111(1460), #112(1460), #113(1460), #114(1460), #115(1460), #116(1460), #117(1460), #118(1460), #119(1460), #120(1460), #121(1460), #122(1460), #123(1460), #124(1460), #125(1460), #126(1460), #127(1460), #128(1460), #129(1460), #130(1460), #131(1460), #132(1460), #133(1460), #134(1460), #135(1460), #136(1460), #137(1460), #138(1460), #139(1460), #140(1460), #141(1460), #142(1460), #143(1460), #144(1460), #145(1460), #146(1460), #147(1460), #148(1460), #149(1460), #150(1460), #151(1460), #152(1460), #153(1460), #154(1460), #155(1460), #156(1460), #157(1460), #158(1460), #159(1460), #160(1460), #161(1460), #162(1460), #163(1460), #164(1460), #165(1460), #166(1460), #167(1460), #168(1460), #169(1460), #170(1460), #171(1460), #172(1460), #173(1460), #174(1460), #175(1460), #176(1460), #177(1460), #178(1460), #179(1460), #180(1460), #181(1460), #182(1460), #183(1460), #184(1460), #185(1460), #186(1460), #187(1460), #188(1460), #189(1460), #190(1460), #191(1460), #192(1460), #193(1460), #194(1460), #195(1460), #196(1460), #197(1460), #198(1460), #199(1460), #200(1460), #201(1460), #202(1460), #203(1460), #204(1460), #205(1460), #206(1460), #207(1460), #208(1460), #209(1460), #210(1460), #211(1460), #212(1460), #213(1460), #214(1460), #215(1460), #216(1460), #217(1460), #218(1460), #219(1460), #220(1460), #221(1460), #222(1460), #223(1460), #224(1460), #225(1460), #226(1460), #227(1460), #228(1460), #229(1460), #230(1460), #231(1460), #232(1460), #233(1460), #234(1460), #235(1460), #236(1460), #237(1460), #238(1460), #239(1460), #240(1460), #241(1460), #242(1460), #243(1460), #244(1460), #245(1460), #246(1460), #247(1460), #248(1460), #249(1460), #250(1460), #251(1460), #252(1460), #253(1460), #254(1460), #255(1460), #256(1460), #257(1460), #258(1460), #259(1460), #260(1460), #261(1460), #262(1460), #263(1460), #264(1460), #265(1460), #266(1460), #267(1460), #268(1460), #269(1460), #270(1460), #271(1460), #272(1460), #273(1460), #274(1460), #275(1460), #276(1460), #277(1460), #278(1460), #279(1460), #280(1460), #281(1460), #282(1460), #283(1460), #284(1460), #285(1460), #286(1460), #287(1460), #288(1460), #289(1460), #290(1460), #291(1460), #292(1460), #293(1460), #294(1460), #295(1460), #296(1460), #297(1460), #298(1460), #299(1460), #300(1460), #301(1460), #302(1460), #303(1460), #304(1460), #305(1460), #306(1460), #307(1460), #308(1460), #309(1460), #310(1460), #311(1460), #312(1460), #313(1460), #314(1460), #315(1460), #316(1460), #317(1460), #318(1460), #319(1460), #320(1460), #321(1460), #322(1460), #323(1460), #324(1460), #325(1460), #326(1460), #327(1460), #328(1460), #329(1460), #330(1460), #331(1460), #332(1460), #333(1460), #334(1460), #335(1460), #336(1460), #337(1460), #338(1460), #339(1460), #340(1460), #341(1460), #342(1460), #343(1460), #344(1460), #345(1460), #346(1460), #347(1460), #348(1460), #349(1460), #350(1460), #351(1460), #352(1460), #353(1460), #354(1460), #355(1460), #356(1460), #357(1460), #358(1460), #359(1460), #360(1460), #361(1460), #362(1460), #363(1460), #364(1460), #365(1460), #366(1460), #367(1460), #368(1460), #369(1460), #370(1460), #371(1460), #372(1460), #373(1460), #374(1460), #375(1460), #376(1460), #377(1460), #378(1460), #379(1460), #380(1460), #381(1460), #382(1460), #383(1460), #384(1460), #385(1460), #386(1460), #387(1460), #388(1460), #389(1460), #390(1460), #391(1460), #392(1460), #393(1460), #394(1460), #395(1460), #396(1460), #397(1460), #398(1460), #399(1460), #400(1460), #401(1460), #402(1460), #403(1460), #404(1460), #405(1460), #406(1460), #407(1460), #408(1460), #409(1460), #410(1460), #411(1460), #412(1460), #413(1460), #414(1460), #415(1460), #416(1460), #417(1460), #418(1460), #419(1460), #420(1460), #421(1460), #422(1460), #423(1460), #424(1460), #425(1460), #426(1460), #427(1460), #428(1460), #429(1460), #430(1460), #431(1460), #432(1460), #433(1460), #434(1460), #435(1460), #436(1460), #437(1460), #438(1460), #439(1460), #440(1460), #441(1460), #442(1460), #443(1460), #444(1460), #445(1460), #446(1460), #447(1460), #448(1460), #449(1460), #450(1460), #451(1460), #452(1460), #453(1460), #454(1460), #455(1460), #456(1460), #457(1460), #458(1460), #459(1460), #460(1460), #461(1460), #462(1460), #463(1460), #464(1460), #465(1460), #466(1460), #467(1460), #468(1460), #469(1460), #470(1460), #471(1460), #472(1460), #473(1460), #474(1460), #475(1460), #476(1460), #477(1460), #478(1460), #479(1460), #480(1460), #481(1460), #482(1460), #483(1460), #484(1460), #485(1460), #486(1460), #487(1460), #488(1460), #489(1460), #490(1460), #491(1460), #492(1460), #493(1460), #494(1460), #495(1460), #496(1460), #497(1460), #498(1460), #499(1460), #500(1460), #501(1460), #502(1460), #503(1460), #504(1460), #505(1460), #506(1460), #507(1460), #508(1460), #509(1460), #510(1460), #511(1460), #512(1460), #513(1460), #514(1460), #515(1460), #516(1460), #517(1460), #518(1460), #519(1460), #520(1460), #521(1460), #522(1460), #523(1460), #524(1460), #525(1460), #526(1460), #527(1460), #528(1460), #529(1460), #530(1460), #531(1460), #532(1460), #533(1460), #534(1460), #535(1460), #536(1460), #537(1460), #538(1460), #539(1460), #540(1460), #541(1460), #542(1460), #543(1460), #544(1460), #545(1460), #546(1460), #547(1460), #548(1460), #549(1460), #550(1460), #551(1460), #552(1460), #553(1460), #554(1460), #555(1460), #556(1460), #557(1460), #558(1460), #559(1460), #560(1460), #561(1460), #562(1460), #563(1460), #564(1460), #565(1460), #566(1460), #567(1460), #568(1460), #569(1460), #570(1460), #571(1460), #572(1460), #573(1460), #574(1460), #575(1460), #576(1460), #577(1460), #578(1460), #579(1460), #580(1460), #581(1460), #582(1460), #583(1460), #584(1460), #585(1460), #586(1460), #587(1460), #588(1460), #589(1460), #590(1460), #591(1460), #592(1460), #593(1460), #594(1460), #595(1460), #596(1460), #597(1460), #598(1460), #599(1460), #600(1460), #601(1460), #602(1460), #603(1460), #604(1460), #605(1460), #606(1460), #607(1460), #608(1460), #609(1460), #610(1460), #611(1460), #612(1460), #613(1460), #614(1460), #615(1460), #616(1460), #617(1460), #618(1460), #619(1460), #620(1460), #621(1460), #622(1460), #623(1460), #624(1460), #625(1460), #626(1460), #627(1460), #628(1460), #629(1460), #630(1460), #631(1460), #632(1460), #633(1460), #634(1460), #635(1460), #636(1460), #637(1460), #638(1460), #639(1460), #640(1460), #641(1460), #642(1460), #643(1460), #644(1460), #645(1460), #646(1460), #647(1460), #648(1460), #649(1460), #650(1460), #651(1460), #652(1460), #653(1460), #654(1460), #655(1460), #656(1460), #657(1460), #658(1460), #659(1460), #660(1460), #661(1460), #662(1460), #663(1460), #664(1460), #665(1460), #666(1460), #667(1460), #668(1460), #669(1460), #670(1460), #671(1460), #672(1460), #673(1460), #674(1460), #675(1460), #676(1460), #677(1460), #678(1460), #679(1460), #680(1460), #681(1460), #682(1460), #683(1460), #684(1460), #685(1460), #686(1460), #687(1460), #688(1460), #689(1460), #690(1460), #691(1460), #692(1460), #693(1460), #694(1460), #695(1460), #696(1460), #697(1460), #698(1460), #699(1460), #700(1460), #701(1460), #702(1460), #703(1460), #704(1460), #705(1460), #706(1460), #707(1460), #708(1460), #709(1460), #710(1460), #711(1460), #712(1460), #713(1460), #714(1460), #715(1460), #716(1460), #717(1460), #718(1460), #719(1460), #720(1460), #721(1460), #722(1460), #723(1460), #724(1460), #725(1460), #726(1460), #727(1460), #728(1460), #729(1460), #730(1460), #731(1460), #732(1460), #733(1460), #734(1460), #735(1460), #736(1460), #737(1460), #738(1460), #739(1460), #740(1460), #741(1460), #742(1460), #743(1460), #744(1460), #745(1460), #746(1460), #747(1460), #748(1460), #749(1460), #750(1460), #751(1460), #752(1460), #753(1460), #754(1460), #755(1460), #756(1460), #757(1460), #758(1460), #759(1460), #760(1460), #761(1460), #762(1460), #763(1460), #764(1460), #765(1460), #766(1460), #767(1460), #768(1460), #769(1460), #770(1460), #771(1460), #772(1460), #773(1460), #774(1460), #775(1460), #776(1460), #777(1460), #778(1460), #779(1460), #780(1460), #781(1460), #782(1460), #783(1460), #784(1460), #785(1460), #786(1460), #787(1460), #788(1460), #789(1460), #790(1460), #791(1460), #792(1460), #793(1460), #794(1460), #795(1460), #796(1460), #797(1460), #798(1460), #799(1460), #800(1460), #801(1460), #802(1460), #803(1460), #804(1460), #805(1460), #806(1460), #807(1460), #808(1460), #809(1460), #810(1460), #811(1460), #812(1460), #813(1460), #814(1460), #815(1460), #816(1460), #817(1460), #818(1460), #819(1460), #820(1460), #821(1460), #822(1460), #823(1460), #824(1460), #825(1460), #826(1460), #827(1460), #828(1460), #829(1460), #830(1460), #831(1460), #832(1460), #833(1460), #834(1460), #835(1460), #836(1460), #837(1460), #838(1460), #839(1460), #840(1460), #841(1460), #842(1460), #843(1460), #844(1460), #845(1460), #846(1460), #847(1460), #848(1460), #849(1460), #850(1460), #851(1460), #852(1460), #853(1460), #854(1460), #855(1460), #856(1460), #857(1460), #858(1460), #859(1460), #860(1460), #861(1460), #862(1460), #863(1460), #864(1460), #865(1460), #866(1460), #867(1460), #868(1460), #869(1460), #870(1460), #871(1460), #872(1460), #873(1460), #874(1460), #875(1460), #876(1460), #877(1460), #878(1460), #879(1460), #880(1460), #881(1460), #882(1460), #883(1460), #884(1460), #885(1460), #886(1460), #887(1460), #888(1460), #889(1460), #890(1460), #891(1460), #892(1460), #893(1460), #894(1460), #895(1460), #896(1460), #897(1460), #898(1460), #899(1460), #900(1460), #901(1460), #902(1460), #903(1460), #904(1460), #905(1460), #906(1460), #907(1460), #908(1460), #909(1460), #910(1460), #911(1460), #912(1460), #913(1460), #914(1460), #915(1460), #916(1460), #917(1460), #918(1460), #919(1460), #920(1460), #921(1460), #922(1460), #923(1460), #924(1460), #925(1460), #926(1460), #927(1460), #928(1460), #929(1460), #930(1460), #931(1460), #932(1460), #933(1460), #934(1460), #935(1460), #936(1460), #937(1460), #938(1460), #939(1460), #940(1460), #941(1460), #942(1460), #943(1460), #944(1460), #945(1460), #946(1460), #947(1460), #948(1460), #949(1460), #950(1460), #951(1460), #952(1460), #953(1460), #954(1460), #955(1460), #956(1460), #957(1460), #958(1460), #959(1460), #960(1460), #961(1460), #962(1460), #963(1460), #964(1460), #965(1460), #966(1460), #967(1460), #968(1460), #969(1460), #970(1460), #971(1460), #972(1460), #973(1460), #974(1460), #975(1460), #976(1460), #977(1460), #978(1460), #979(1460), #980(1460), #981(1460), #982(1460), #983(1460), #984(1460), #985(1460), #986(1460), #987(1460), #988(1460), #989(1460), #990(1460), #991(1460), #992(1460), #993(1460), #994(1460), #995(1460), #996(1460), #997(1460), #998(1460), #999(1460), #1000(1460), #1001(1460), #1002(1460), #1003(1460), #1004(1460), #1005(1460), #1006(1460), #1007(1460), #1008(1460), #1009(1460), #1010(1460), #1011(1460), #1012(1460), #1013(1460), #1014(1460), #1015(1460), #1016(1460), #1017(1460), #1018(1460), #1019(1460), #1020(1460), #1021(1460), #1022(1460), #1023(1460), #1024(1460), #1025(1460), #1026(1460), #1027(1460), #1028(1460), #1029(1460), #1030(1460), #1031(1460), #1032(1460), #1033(1460), #1034(1460), #1035(1460), #1036(1460), #1037(1460), #1038(1460), #1039(1460), #1040(1460), #1041(1460), #1042(1460), #1043(1460), #1044(1460), #1045(1460), #1046(1460), #1047(1460), #1048(1460), #1049(1460), #1050(1460), #1051(1460), #1052(1460), #1053(1460), #1054(1460), #1055(1460), #1056(1460), #1057(1460), #1058(1460), #1059(1460), #1060(1460), #1061(1460), #1062(1460), #1063(1460), #1064(1460), #1065(1460), #1066(1460), #1067(1460), #1068(1460), #1069(1460), #1070(1460), #1071(1460), #1072(1460), #1073(1460), #1074(1460), #1075(1460), #1076(1460), #1077(1460), #1078(1460), #1079(1460), #1080(1460), #1081(1460), #1082(1460), #1083(1460), #1084(1460), #1085(1460), #1086(1460), #1087(1460), #1088(1460), #1089(1460), #1090(1460), #1091(1460), #1092(1460), #1093(1460), #1094(1460), #1095(1460), #1096(1460

**Figura 73.** Fluxos de tráfego multimídia de áudio nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, e seus respectivos identificadores DSCP no ambiente de emulação.

**(A)**

Filter: `udp.port==7001` Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
8	40.058463	200.200.200.2	172.18.0.4	UDP	105	Source port: 60093 Destination port: 7001
10	40.108826	200.200.200.2	172.18.0.4	UDP	105	Source port: 51788 Destination port: 7001
16	40.175335	200.200.200.2	172.18.0.4	UDP	105	Source port: 60844 Destination port: 7001
22	40.216237	200.200.200.2	172.18.0.4	UDP	105	Source port: 38770 Destination port: 7001
33	40.266288	200.200.200.2	172.18.0.4	UDP	105	Source port: 40254 Destination port: 7001
49	40.322984	200.200.200.2	172.18.0.4	UDP	105	Source port: 46732 Destination port: 7001

▶ Frame 10: 105 bytes on wire (840 bits), 105 bytes captured (840 bits)  
 ▶ Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:85:dc:eb (08:00:27:85:dc:eb)  
 ▼ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.4 (172.18.0.4)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0x44 (DSCP 0x11: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0100 01.. = Differentiated Services Codepoint: Unknown (0x11)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 91  
 Identification: 0x447f (17535)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 62  
 Protocol: UDP (17)  
 ▶ Header checksum: 0xbaed [correct]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 172.18.0.4 (172.18.0.4)  
 ▶ User Datagram Protocol, Src Port: 51788 (51788), Dst Port: 7001 (7001)  
 ▶ Data (63 bytes)

**(B)**

Filter: `udp.port==7002` Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
8	27.716589	200.200.200.2	172.18.0.5	UDP	105	Source port: 33925 Destination port: 7002
14	27.797462	200.200.200.2	172.18.0.5	UDP	105	Source port: 38033 Destination port: 7002
25	27.855014	200.200.200.2	172.18.0.5	UDP	105	Source port: 35580 Destination port: 7002
35	27.923482	200.200.200.2	172.18.0.5	UDP	105	Source port: 50870 Destination port: 7002
39	27.946000	200.200.200.2	172.18.0.5	UDP	105	Source port: 33521 Destination port: 7002
46	27.985715	200.200.200.2	172.18.0.5	UDP	105	Source port: 56425 Destination port: 7002

▶ Frame 14: 105 bytes on wire (840 bits), 105 bytes captured (840 bits)  
 ▶ Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:c6:13:07 (08:00:27:c6:13:07)  
 ▼ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.5 (172.18.0.5)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0x48 (DSCP 0x12: Assured Forwarding 21; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0100 10.. = Differentiated Services Codepoint: Assured Forwarding 21 (0x12)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 91  
 Identification: 0x2dd0 (11728)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 62  
 Protocol: UDP (17)  
 ▶ Header checksum: 0xd197 [correct]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 172.18.0.5 (172.18.0.5)  
 ▶ User Datagram Protocol, Src Port: 38033 (38033), Dst Port: 7002 (7002)  
 ▶ Data (63 bytes)

**Figura 74.** Fluxos de tráfego multimídia de áudio sobre RTSP (A) e sobre RTP (B), pertinentes ao serviço de *streaming* de áudio, e seus respectivos identificadores DSCP no ambiente de emulação.

**(A)**

No.	Time	Source	Destination	Protocol	Length	Info
4	0.009049000	172.18.0.3	200.200.200.1	RTSP	454	DESCRIBE rtsp://200.200.200.1/sample.mp3 RTSP/1.0
6	0.067065000	200.200.200.1	172.18.0.3	RTSP/SDP	880	Reply: RTSP/1.0 200 OK, with session description
8	0.069290000	172.18.0.3	200.200.200.1	RTSP	414	SETUP rtsp://200.200.200.1/sample.mp3/rtx RTSP/1.0
9	0.081949000	200.200.200.1	172.18.0.3	RTSP	631	Reply: RTSP/1.0 200 OK
10	0.082171000	172.18.0.3	200.200.200.1	RTSP	483	SET_PARAMETER rtsp://200.200.200.1/sample.mp3 RTSP/1.0
11	0.097359000	200.200.200.1	172.18.0.3	RTSP	302	Reply: RTSP/1.0 200 OK (application/x-rtsp-udp-packetp

Frame 9: 631 bytes on wire (5048 bits), 631 bytes captured (5048 bits)

Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:8a:2a:6a (08:00:27:8a:2a:6a)

Internet Protocol Version 4, Src: 200.200.200.1 (200.200.200.1), Dst: 172.18.0.3 (172.18.0.3)

Version: 4  
Header Length: 20 bytes

Differentiated Services Field: 0x4c (DSCP 0x13: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

0100 11.. = Differentiated Services Codepoint: Unknown (0x13)

.... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)

Total Length: 617  
Identification: 0x773c (30524)

Flags: 0x02 (Don't Fragment)  
Fragment offset: 0  
Time to live: 126  
Protocol: TCP (6)

Header checksum: 0x4627 [correct]  
Source: 200.200.200.1 (200.200.200.1)  
Destination: 172.18.0.3 (172.18.0.3)

Transmission Control Protocol, Src Port: 554 (554), Dst Port: 1039 (1039), Seq: 2287, Ack: 761, Len: 577

Real Time Streaming Protocol

**(B)**

No.	Time	Source	Destination	Protocol	Length	Info
6	0.067065000	200.200.200.1	172.18.0.3	RTSP/SDP	880	Reply: RTSP/1.0 200 OK, with session description
8	0.069290000	172.18.0.3	200.200.200.1	RTSP	414	SETUP rtsp://200.200.200.1/sample.mp3/rtx RTSP/1.0
9	0.081949000	200.200.200.1	172.18.0.3	RTSP	631	Reply: RTSP/1.0 200 OK
10	0.082171000	172.18.0.3	200.200.200.1	RTSP	483	SET_PARAMETER rtsp://200.200.200.1/sample.mp3 RTSP/1.0
11	0.097359000	200.200.200.1	172.18.0.3	RTSP	302	Reply: RTSP/1.0 200 OK (application/x-rtsp-udp-packetp
12	0.097386000	200.200.200.1	172.18.0.3	RTP	1512	PT=DynamicRTP-Type-120, SSRC=0x863D09A, Seq=45091, Tim

Frame 12: 1512 bytes on wire (12096 bits), 1512 bytes captured (12096 bits)

Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:8a:2a:6a (08:00:27:8a:2a:6a)

Internet Protocol Version 4, Src: 200.200.200.1 (200.200.200.1), Dst: 172.18.0.3 (172.18.0.3)

Version: 4  
Header Length: 20 bytes

Differentiated Services Field: 0x4c (DSCP 0x13: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

0100 11.. = Differentiated Services Codepoint: Unknown (0x13)

.... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)

Total Length: 1498  
Identification: 0x773d (30525)

Flags: 0x00  
Fragment offset: 0  
Time to live: 126  
Protocol: UDP (17)

Header checksum: 0x82aa [correct]  
Source: 200.200.200.1 (200.200.200.1)  
Destination: 172.18.0.3 (172.18.0.3)

User Datagram Protocol, Src Port: 5004 (5004), Dst Port: 1040 (1040)

Real-Time Transport Protocol

**Figura 75.** Fluxos de tráfego multimídia de voz nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, e seus respectivos identificadores DSCP no ambiente de emulação.

**(A)**

Filter: `udp.port==8001` Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
9	40.059172	200.200.200.2	172.18.0.4	UDP	95	Source port: 44458 Destination port: 8001
12	40.108844	200.200.200.2	172.18.0.4	UDP	95	Source port: 45702 Destination port: 8001
13	40.136749	200.200.200.2	172.18.0.4	UDP	95	Source port: 34327 Destination port: 8001
26	40.229124	200.200.200.2	172.18.0.4	UDP	95	Source port: 55342 Destination port: 8001
36	40.273228	200.200.200.2	172.18.0.4	UDP	95	Source port: 59805 Destination port: 8001
50	40.323014	200.200.200.2	172.18.0.4	UDP	95	Source port: 43190 Destination port: 8001

Frame 12: 95 bytes on wire (760 bits), 95 bytes captured (760 bits)

Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:85:dc:eb (08:00:27:85:dc:eb)

Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.4 (172.18.0.4)

Version: 4  
Header length: 20 bytes

Differentiated Services Field: 0x84 (DSCP 0x21: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

1000 01.. = Differentiated Services Codepoint: Unknown (0x21)

.... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)

Total Length: 81  
Identification: 0x447d (17533)

Flags: 0x02 (Don't Fragment)  
Fragment offset: 0  
Time to live: 62  
Protocol: UDP (17)

Header checksum: 0xbab9 [correct]  
Source: 200.200.200.2 (200.200.200.2)  
Destination: 172.18.0.4 (172.18.0.4)

User Datagram Protocol, Src Port: 45702 (45702), Dst Port: 8001 (8001)

Data (53 bytes)

**(B)**

Filter: `udp.port==8002` Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
5	27.692673	200.200.200.2	172.18.0.5	UDP	95	Source port: 41136 Destination port: 8002
13	27.791079	200.200.200.2	172.18.0.5	UDP	95	Source port: 53895 Destination port: 8002
20	27.847832	200.200.200.2	172.18.0.5	UDP	95	Source port: 44110 Destination port: 8002
37	27.927203	200.200.200.2	172.18.0.5	UDP	95	Source port: 46122 Destination port: 8002
44	27.972328	200.200.200.2	172.18.0.5	UDP	95	Source port: 45496 Destination port: 8002
48	28.022039	200.200.200.2	172.18.0.5	UDP	95	Source port: 51914 Destination port: 8002

Frame 13: 95 bytes on wire (760 bits), 95 bytes captured (760 bits)

Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:c6:13:07 (08:00:27:c6:13:07)

Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.5 (172.18.0.5)

Version: 4  
Header length: 20 bytes

Differentiated Services Field: 0x88 (DSCP 0x22: Assured Forwarding 41; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

1000 10.. = Differentiated Services Codepoint: Assured Forwarding 41 (0x22)

.... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)

Total Length: 81  
Identification: 0x2dcf (11727)

Flags: 0x02 (Don't Fragment)  
Fragment offset: 0  
Time to live: 62  
Protocol: UDP (17)

Header checksum: 0xd162 [correct]  
Source: 200.200.200.2 (200.200.200.2)  
Destination: 172.18.0.5 (172.18.0.5)

User Datagram Protocol, Src Port: 53895 (53895), Dst Port: 8002 (8002)

Data (53 bytes)

**Figura 76.** Fluxos de tráfego multimídia de vídeo nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, e seus respectivos identificadores DSCP no ambiente de emulação.

**(A)**

No.	Time	Source	Destination	Protocol	Length	Info
7	40.056537	200.200.200.2	172.18.0.4	UDP	135	Source port: 37660 Destination port: 9001
11	40.108841	200.200.200.2	172.18.0.4	UDP	135	Source port: 35719 Destination port: 9001
17	40.175355	200.200.200.2	172.18.0.4	UDP	135	Source port: 41123 Destination port: 9001
21	40.215916	200.200.200.2	172.18.0.4	UDP	135	Source port: 54863 Destination port: 9001
32	40.266129	200.200.200.2	172.18.0.4	UDP	135	Source port: 49882 Destination port: 9001
51	40.327158	200.200.200.2	172.18.0.4	UDP	135	Source port: 38846 Destination port: 9001

Filter: `udp.port==9001` Expression... Clear Apply

Frame 11: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits)

Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:85:dc:eb (08:00:27:85:dc:eb)

Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.4 (172.18.0.4)

Version: 4  
Header length: 20 bytes

Differentiated Services Field: 0xc4 (DSCP 0x31: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

**1100 01.. = Differentiated Services Codepoint: Unknown (0x31)**

... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)

Total Length: 121  
Identification: 0x447e (17534)

Flags: 0x02 (Don't Fragment)  
Fragment offset: 0  
Time to live: 62  
Protocol: UDP (17)

Header checksum: 0xba50 [correct]  
Source: 200.200.200.2 (200.200.200.2)  
Destination: 172.18.0.4 (172.18.0.4)

User Datagram Protocol, Src Port: 35719 (35719), Dst Port: 9001 (9001)

Data (93 bytes)

**(B)**

No.	Time	Source	Destination	Protocol	Length	Info
9	27.719227	200.200.200.2	172.18.0.5	UDP	135	Source port: 47842 Destination port: 9002
18	27.815447	200.200.200.2	172.18.0.5	UDP	135	Source port: 53514 Destination port: 9002
36	27.923493	200.200.200.2	172.18.0.5	UDP	135	Source port: 43376 Destination port: 9002
45	27.972350	200.200.200.2	172.18.0.5	UDP	135	Source port: 48176 Destination port: 9002
49	28.022716	200.200.200.2	172.18.0.5	UDP	135	Source port: 34600 Destination port: 9002
53	28.096239	200.200.200.2	172.18.0.5	UDP	135	Source port: 45093 Destination port: 9002

Filter: `udp.port==9002` Expression... Clear Apply

Frame 18: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits)

Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:c6:13:07 (08:00:27:c6:13:07)

Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.5 (172.18.0.5)

Version: 4  
Header length: 20 bytes

Differentiated Services Field: 0xc8 (DSCP 0x32: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))

**1100 10.. = Differentiated Services Codepoint: Unknown (0x32)**

... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)

Total Length: 121  
Identification: 0x2dd1 (11729)

Flags: 0x02 (Don't Fragment)  
Fragment offset: 0  
Time to live: 62  
Protocol: UDP (17)

Header checksum: 0xd0f8 [correct]  
Source: 200.200.200.2 (200.200.200.2)  
Destination: 172.18.0.5 (172.18.0.5)

User Datagram Protocol, Src Port: 53514 (53514), Dst Port: 9002 (9002)

Data (93 bytes)

**Figura 77.** Fluxos de tráfego multimídia de vídeo sobre RTSP (A) e sobre RTP (B), pertinentes ao serviço de *streaming* de vídeo, e seus respectivos identificadores DSCP no ambiente de emulação.

**(A)**

No.	Time	Source	Destination	Protocol	Length	Info
4	0.055471000	172.18.0.1	200.200.200.1	RTSP	454	DESCRIBE rtsp://200.200.200.1/sample.wmv RTSP/1.0
15	0.444429000	200.200.200.1	172.18.0.1	RTSP/SDP	182	Reply: RTSP/1.0 200 OK, with session description
16	0.446530000	172.18.0.1	200.200.200.1	RTSP	414	SETUP rtsp://200.200.200.1/sample.wmv/rtx RTSP/1.0
17	0.502832000	200.200.200.1	172.18.0.1	RTSP	631	Reply: RTSP/1.0 200 OK
18	0.503100000	172.18.0.1	200.200.200.1	RTSP	483	SET_PARAMETER rtsp://200.200.200.1/sample.wmv RTSP/1.0
19	0.556983000	200.200.200.1	172.18.0.1	RTP	1512	PT=DynamicRTP-Type-110, SSRC=0xCF0A5339, Seq=35196, Ti

Filter: **rtsp or rtsp** Expression... Clear Apply  
 ▶ Frame 17: 631 bytes on wire (5048 bits), 631 bytes captured (5048 bits)  
 ▶ Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:ad:53:a6 (08:00:27:ad:53:a6)  
 ▶ Internet Protocol Version 4, Src: 200.200.200.1 (200.200.200.1), Dst: 172.18.0.1 (172.18.0.1)  
 Version: 4  
 Header Length: 20 bytes  
 ▶ Differentiated Services Field: 0xcc (DSCP 0x33: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 1100 11.. = Differentiated Services Codepoint: Unknown (0x33)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 617  
 Identification: 0x75f7 (30199)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 126  
 Protocol: TCP (6)  
 ▶ Header checksum: 0x46ee [correct]  
 Source: 200.200.200.1 (200.200.200.1)  
 Destination: 172.18.0.1 (172.18.0.1)  
 ▶ Transmission Control Protocol, Src Port: 554 (554), Dst Port: 1035 (1035), Seq: 8889, Ack: 761, Len: 577  
 ▶ Real Time Streaming Protocol

**(B)**

No.	Time	Source	Destination	Protocol	Length	Info
4	0.055471000	172.18.0.1	200.200.200.1	RTSP	454	DESCRIBE rtsp://200.200.200.1/sample.wmv RTSP/1.0
15	0.444429000	200.200.200.1	172.18.0.1	RTSP/SDP	182	Reply: RTSP/1.0 200 OK, with session description
16	0.446530000	172.18.0.1	200.200.200.1	RTSP	414	SETUP rtsp://200.200.200.1/sample.wmv/rtx RTSP/1.0
17	0.502832000	200.200.200.1	172.18.0.1	RTSP	631	Reply: RTSP/1.0 200 OK
18	0.503100000	172.18.0.1	200.200.200.1	RTSP	483	SET_PARAMETER rtsp://200.200.200.1/sample.wmv RTSP/1.0
19	0.556983000	200.200.200.1	172.18.0.1	RTP	1512	PT=DynamicRTP-Type-110, SSRC=0xCF0A5339, Seq=35196, Ti

Filter: **rtsp or rtsp** Expression... Clear Apply  
 ▶ Frame 19: 1512 bytes on wire (12096 bits), 1512 bytes captured (12096 bits)  
 ▶ Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:ad:53:a6 (08:00:27:ad:53:a6)  
 ▶ Internet Protocol Version 4, Src: 200.200.200.1 (200.200.200.1), Dst: 172.18.0.1 (172.18.0.1)  
 Version: 4  
 Header Length: 20 bytes  
 ▶ Differentiated Services Field: 0xcc (DSCP 0x33: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 1100 11.. = Differentiated Services Codepoint: Unknown (0x33)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 1498  
 Identification: 0x7628 (30248)  
 ▶ Flags: 0x00  
 Fragment offset: 0  
 Time to live: 126  
 Protocol: UDP (17)  
 ▶ Header checksum: 0x8341 [correct]  
 Source: 200.200.200.1 (200.200.200.1)  
 Destination: 172.18.0.1 (172.18.0.1)  
 ▶ User Datagram Protocol, Src Port: 5004 (5004), Dst Port: 1036 (1036)  
 ▶ Real-Time Transport Protocol

## A.2 – Identificação e Encaminhamento Diferenciado de Fluxos de Tráfego Multimídia

De modo complementar à Seção 4.2.2, esta Seção discorre sobre as regras de fluxos empregadas para viabilizar a identificação e o encaminhamento diferenciado de todos os fluxos de tráfego multimídia experimentados no ambiente de emulação.

Nesse contexto, como exemplo, as Figura 78 e 79, que são pertinentes a listagem de regras de fluxos junto ao controlador OpenFlow, ilustram as definições empregadas para viabilizar o encaminhamento diferenciado dos fluxos de dados pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação.

**Figura 78.** Definições empregadas para viabilizar o encaminhamento diferenciado dos fluxos de dados pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação.

```

    "redir-data-1": {
      [...]
      "instructions": {
        "instruction_apply_actions": {
          "actions": "set_field=ipv4_dst->172.18.0.4,
set_field=tcp_dst->6001,output=normal"
        }
      },
      "match": {
        "eth_type": "0x800",
        "ip_dscp": "0x1",
        "ip_proto": "0x6"
      },
      [...]
      "version": "OF_13"
    }

    "redir-data-2": {
      [...]
      "instructions": {
        "instruction_apply_actions": {
          "actions": "set_field=ipv4_dst->172.18.0.5,
set_field=tcp_dst->6002,output=normal"
        }
      },
      "match": {
        "eth_type": "0x800",
        "ip_dscp": "0x2",
        "ip_proto": "0x6"
      },
      [...]
      "version": "OF_13"
    }
  
```



**Figura 79.** Definições complementares empregadas para viabilizar o encaminhamento diferenciado dos fluxos de dados pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação.

```

        "resp-redirect-data-1": {
            [...]
            "instructions": {
                "instruction_apply_actions": {
                    "actions": "set_field=ipv4_src->188.188.188.1,
set_field=tcp_src->6111,output=normal"
                }
            },
            "match": {
                "eth_type": "0x800",
                "ip_proto": "0x6",
                "ipv4_src": "172.18.0.4",
                "tcp_src": "6001"
            },
            [...]
            "version": "OF_13"
        }

        "resp-redirect-data-2": {
            [...]
            "instructions": {
                "instruction_apply_actions": {
                    "actions": "set_field=ipv4_src->188.188.188.1,
set_field=tcp_src->6112,output=normal"
                }
            },
            "match": {
                "eth_type": "0x800",
                "ip_proto": "0x6",
                "ipv4_src": "172.18.0.5",
                "tcp_src": "6002"
            },
            [...]
            "version": "OF_13"
        }
    
```

Nesse contexto, no que se refere à Figura 78, a regra de fluxo “*redirect-data-1*” é pertinente ao encaminhamento diferenciado para os fluxos de dados no canal 1 e a regra de fluxo “*redirect-data-2*” é pertinente ao encaminhamento diferenciado para os fluxos de dados no canal 2, ambos quanto aos fluxos gerados por meio do gerador de tráfego multimídia. Por sua vez, em ambas as regras de fluxos, instruções distintas de encaminhamento são definidas de acordo com seus respectivos *match fields* OpenFlow. Nesse caso, na regra de fluxo “*redirect-data-1*”, a instrução é direcionar os fluxos em questão para o cliente multimídia H6 na porta 6001/TCP. Para tal, essa regra é aplicada aos fluxos IPv4 com identificador DSCP 0x01 e que utilizam o protocolo TCP como protocolo de transporte. Por sua vez, na regra de fluxo “*redirect-*

*data-2*”, a instrução é direcionar os fluxos em questão para o cliente multimídia H7 na porta 6002/TCP. Para tal, essa regra é aplicada aos fluxos IPv4 com identificador DSCP 0x02 e que também utilizam o protocolo TCP como protocolo de transporte. Em ambas as regras, a versão empregada do protocolo OpenFlow é a 1.3.

Por sua vez, quanto à Figura 79, a regra de fluxo “*resp-redirect-data-1*” é pertinente às definições complementares para o encaminhamento diferenciado para os fluxos de dados no canal 1 e a regra de fluxo “*resp-redirect-data-2*” é pertinente às definições complementares pertinentes ao encaminhamento diferenciado para os fluxos de dados no canal 2. Por sua vez, em ambas as regras de fluxos, instruções distintas de encaminhamento são definidas de acordo com seus respectivos *match fields* OpenFlow. Nesse caso, na regra de fluxo “*resp-redirect-data-1*”, a instrução é direcionar as respostas pertinentes aos fluxos em questão como originadas pelo *gateway* multimídia em sua porta 6111/TCP (“*actions*”: “*set\_field=ipv4\_src->188.188.188.1, set\_field=tcp\_src->6111,output=normal*”). Para tal, essa regra é aplicada aos fluxos IPv4 (“*eth\_type*”: “*0x800*”), que utilizam o protocolo TCP (“*ip\_proto*”: “*0x6*”) como protocolo de transporte, originados pelo cliente multimídia H6 (“*ipv4\_src*”: “*172.18.0.4*”) em sua porta 6001/TCP (“*tcp\_src*”: “*6001*”). Por sua vez, na regra de fluxo “*resp-redirect-data-2*”, a instrução é direcionar as respostas pertinentes aos fluxos em questão como originadas pelo *gateway* multimídia em sua porta 6112/TCP. Para tal, essa regra é aplicada aos fluxos IPv4, que também utilizam o protocolo TCP como protocolo de transporte, porém originados pelo cliente multimídia H7 em sua porta 6002/TCP.

Em tais exemplos, assim como realizado na Seção anterior, é importante destacar que apenas as informações pertinentes aos nomes das regras de fluxos e suas respectivas definições em termos de instruções (ações) e *match fields*, bem como quanto a versão empregada do protocolo OpenFlow, são ilustradas em tal figura. As demais informações, tidas como não relevantes para os exemplos em questão, foram substituídas e ilustradas pela *string* “[...]”.

Nesse cenário, quanto aos fluxos de áudio pertinentes aos canais 1 e 2 do gerador de tráfego multimídia, a Figura 80 ilustra as definições empregadas para viabilizar o encaminhamento diferenciado de ambos no ambiente de emulação. Em tal exemplo, a regra de fluxo “*redirect-audio-1*” é pertinente ao encaminhamento diferenciado para os fluxos de áudio no canal 1 e a regra de fluxo “*redirect-audio-2*” é pertinente ao encaminhamento diferenciado para os fluxos de áudio no canal 2, ambos quanto aos fluxos gerados por meio do gerador de tráfego multimídia. Por sua vez, em ambas as regras de fluxos, instruções distintas

de encaminhamento são definidas de acordo com seus respectivos *match fields* OpenFlow. Nesse caso, na regra de fluxo “*redir-audio-1*”, a instrução é direcionar os fluxos em questão para o cliente multimídia H6 na porta 7001/UDP. Para tal, essa regra é aplicada aos fluxos IPv4 com identificador DSCP 0x11 e que utilizam o protocolo UDP como protocolo de transporte. Por sua vez, na regra de fluxo “*redir-audio-2*”, a instrução é direcionar os fluxos em questão para o cliente multimídia H7 na porta 7002/UDP. Para tal, essa regra é aplicada aos fluxos IPv4 com identificador DSCP 0x12 e que também utilizam o protocolo UDP como protocolo de transporte. Em ambas as regras, a versão empregada do protocolo OpenFlow é a 1.3.

**Figura 80.** Definições empregadas para viabilizar o encaminhamento diferenciado dos fluxos de áudio pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação.

```

    "redir-audio-1": {
      [...]
      "instructions": {
        "instruction_apply_actions": {
          "actions": "set_field=ipv4_dst->172.18.0.4,
set_field=udp_dst->7001,output=normal"
        }
      },
      "match": {
        "eth_type": "0x800",
        "ip_dscp": "0x11",
        "ip_proto": "0x11"
      },
      [...]
      "version": "OF_13"
    }

    "redir-audio-2": {
      [...]
      "instructions": {
        "instruction_apply_actions": {
          "actions": "set_field=ipv4_dst->172.18.0.5,
set_field=udp_dst->7002,output=normal"
        }
      },
      "match": {
        "eth_type": "0x800",
        "ip_dscp": "0x12",
        "ip_proto": "0x11"
      },
      [...]
      "version": "OF_13"
    }

```

No que se refere aos fluxos de voz pertinentes aos canais 1 e 2 do gerador de tráfego multimídia, a Figura 81 ilustra as definições empregadas para viabilizar o encaminhamento diferenciado de ambos no ambiente de emulação. Nesse caso, na regra de fluxo “*redir-voice-1*”, que é pertinente ao encaminhamento diferenciado para os fluxos de voz no canal 1, a instrução é direcionar os fluxos em questão para o cliente multimídia H6 na porta 8001/UDP. Para tal, essa regra é aplicada aos fluxos IPv4 com identificador DSCP 0x21 e que utilizam o protocolo UDP como protocolo de transporte. Por sua vez, na regra de fluxo “*redir-voice-2*”, que é pertinente ao encaminhamento diferenciado para os fluxos de voz no canal 2, a instrução é direcionar os fluxos em questão para o cliente multimídia H7 na porta 8002/UDP. Para tal, essa regra é aplicada aos fluxos IPv4 com identificador DSCP 0x22 e que também utilizam o protocolo UDP como protocolo de transporte.

**Figura 81.** Definições empregadas para viabilizar o encaminhamento diferenciado dos fluxos de voz pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação.

```

"redir-voice-1": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ipv4_dst->172.18.0.4,
set_field=udp_dst->8001,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_dscp": "0x21",
    "ip_proto": "0x11"
  },
  [...]
  "version": "OF_13"
}

"redir-voice-2": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ipv4_dst->172.18.0.5,
set_field=udp_dst->8002,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_dscp": "0x22",
    "ip_proto": "0x11"
  },
  [...]
  "version": "OF_13"
}

```

Quanto aos fluxos de vídeo pertinentes aos canais 1 e 2 do gerador de tráfego multimídia, a Figura 82 ilustra as definições empregadas para viabilizar o encaminhamento diferenciado de ambos no ambiente de emulação. Nesse caso, na regra de fluxo “*redir-video-1*”, que é pertinente ao encaminhamento diferenciado para os fluxos de vídeo no canal 1, a instrução é direcionar os fluxos em questão para o cliente multimídia H6 na porta 9001/UDP. Para tal, essa regra é aplicada aos fluxos IPv4 com identificador DSCP 0x31 e que utilizam o protocolo UDP como protocolo de transporte. Por sua vez, na regra de fluxo “*redir-video-2*”, que é pertinente ao encaminhamento diferenciado para os fluxos de vídeo no canal 2, a instrução é direcionar os fluxos em questão para o cliente multimídia H7 na porta 9002/UDP. Para tal, essa regra é aplicada aos fluxos IPv4 com identificador DSCP 0x32 e que também utilizam o protocolo UDP como protocolo de transporte.

**Figura 82.** Definições empregadas para viabilizar o encaminhamento diferenciado dos fluxos de vídeo pertinentes aos canais 1 e 2 do gerador de tráfego multimídia no ambiente de emulação.

```

"redir-video-1": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ipv4_dst->172.18.0.4,
set_field=udp_dst->9001,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_dscp": "0x31",
    "ip_proto": "0x11"
  },
  [...]
  "version": "OF_13"
}

"redir-video-2": {
  [...]
  "instructions": {
    "instruction_apply_actions": {
      "actions": "set_field=ipv4_dst->172.18.0.5,
set_field=udp_dst->9002,output=normal"
    }
  },
  "match": {
    "eth_type": "0x800",
    "ip_dscp": "0x32",
    "ip_proto": "0x11"
  },
  [...]
  "version": "OF_13"
}

```

Diante desse cenário, durante a realização das avaliações experimentais, a identificação e o encaminhamento diferenciado de fluxos de tráfego multimídia, nesse caso, por meio de identificadores DSCP específicos, mostraram-se funcionais no ambiente de emulação. Em linhas gerais, houve um processo de comunicação entre as aplicações transmissoras e receptoras dos conteúdos multimídia de áudio, de voz, de vídeo e de dados no ambiente SDN. Por sua vez, com a alteração e/ou remoção de qualquer uma das regras de fluxos pertinentes aos conteúdos multimídia em questão, esse processo de comunicação era imediatamente percebido no ambiente de emulação, com o(s) fluxo(s) em questão deixando de ser encaminhado(s) ao(s) seu(s) respectivo(s) cliente(s) multimídia.

Nesse contexto, como exemplo, as Figuras 83 e 84 ilustram os fluxos multimídia de dados recebidos e encaminhados, respectivamente, pelo *gateway* multimídia aos clientes multimídia H6 e H7; as Figuras 85 e 86 ilustram os fluxos multimídia de áudio recebidos e encaminhados, respectivamente, pelo *gateway* multimídia aos clientes multimídia H6 e H7; as Figuras 87 e 88 ilustram os fluxos multimídia de voz recebidos e encaminhados, respectivamente, pelo *gateway* multimídia aos clientes multimídia H6 e H7; as Figuras 89 e 90 ilustram os fluxos multimídia de vídeo recebidos e encaminhados, respectivamente, pelo *gateway* multimídia aos clientes multimídia H6 e H7; e, por fim, a Figura 91 ilustra o recebimento de ambos os fluxos multimídia de dados, de áudio, de voz e de vídeo pelos clientes multimídia H6 e H7.

Em tais ilustrações, observe que os fluxos recebidos pelo *gateway* multimídia (Figuras 83, 85, 87 e 89) possuem seu endereço IP como destinatário, bem como as portas de comunicação especificadas anteriormente junto à aplicação transmissora do gerador de tráfego multimídia. Por outro lado, os fluxos encaminhados pelo *gateway* multimídia (Figuras 84, 86, 88 e 90) possuem os endereços IP dos clientes multimídia H6 e H7 como destinatários, bem como suas respectivas portas de escuta no âmbito da aplicação receptora do gerador de tráfego multimídia. De modo complementar, observe que, nos clientes multimídia (Figura 91), todos os fluxos recebidos possuem o endereço IP do servidor multimídia H2 enquanto sistema transmissor, de modo que, em linhas gerais, o *gateway* multimídia foi tido como “transparente” durante esse processo de encaminhamento diferenciado de fluxos de tráfego multimídia no ambiente de emulação.

**Figura 83.** Exemplo de recebimento de fluxos multimídia de dados nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao *gateway* multimídia no ambiente de emulação.

**(A)**

No.	Time	Source	Destination	Protocol	Length	Info
38	6.072394000	200.200.200.2	188.188.188.1	TCP	74	51295 > 6111 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SAC
40	6.080523000	188.188.188.1	200.200.200.2	TCP	74	6111 > 51295 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 M
42	6.088223000	200.200.200.2	188.188.188.1	TCP	66	51295 > 6111 [ACK] Seq=1 Ack=1 Win=29248 Len=0 TSval=
44	6.098135000	200.200.200.2	188.188.188.1	TCP	691	51295 > 6111 [PSH, ACK] Seq=1 Ack=1 Win=29248 Len=625
45	6.101744000	200.200.200.2	188.188.188.1	TCP	66	51295 > 6111 [FIN, ACK] Seq=626 Ack=1 Win=29248 Len=6
46	6.110812000	188.188.188.1	200.200.200.2	TCP	66	6111 > 51295 [ACK] Seq=1 Ack=626 Win=30272 Len=0 TSva

▶ Frame 38: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 1  
 ▶ Ethernet II, Src: 08:00:27:d4:00:e9 (08:00:27:d4:00:e9), Dst: 08:00:27:2b:f2:57 (08:00:27:2b:f2:57)  
 ▼ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 188.188.188.1 (188.188.188.1)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0x04 (DSCP 0x01: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0000 01.. = Differentiated Services Codepoint: Unknown (0x01)  
 .... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 60  
 Identification: 0xb215 (45589)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 63  
 Protocol: TCP (6)  
 ▶ Header checksum: 0x8019 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 188.188.188.1 (188.188.188.1)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 ▶ Transmission Control Protocol, Src Port: 51295 (51295), Dst Port: 6111 (6111), Seq: 0, Len: 0

**(B)**

No.	Time	Source	Destination	Protocol	Length	Info
337	7.845502000	200.200.200.2	188.188.188.1	TCP	74	49942 > 6112 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SAC
339	7.863650000	188.188.188.1	200.200.200.2	TCP	74	6112 > 49942 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 M
340	7.875704000	200.200.200.2	188.188.188.1	TCP	66	49942 > 6112 [ACK] Seq=1 Ack=1 Win=29248 Len=0 TSval=
341	7.875723000	200.200.200.2	188.188.188.1	TCP	691	49942 > 6112 [PSH, ACK] Seq=1 Ack=1 Win=29248 Len=625
342	7.875731000	200.200.200.2	188.188.188.1	TCP	66	49942 > 6112 [FIN, ACK] Seq=626 Ack=1 Win=29248 Len=6
343	7.884458000	200.200.200.2	188.188.188.1	TCP	74	49942 > 6112 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SAC

▶ Frame 337: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 1  
 ▶ Ethernet II, Src: 08:00:27:d4:00:e9 (08:00:27:d4:00:e9), Dst: 08:00:27:2b:f2:57 (08:00:27:2b:f2:57)  
 ▼ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 188.188.188.1 (188.188.188.1)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0x08 (DSCP 0x02: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0000 10.. = Differentiated Services Codepoint: Unknown (0x02)  
 .... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 60  
 Identification: 0x7abc (31420)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 63  
 Protocol: TCP (6)  
 ▶ Header checksum: 0xb76e [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 188.188.188.1 (188.188.188.1)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 ▶ Transmission Control Protocol, Src Port: 49942 (49942), Dst Port: 6112 (6112), Seq: 0, Len: 0

**Figura 84.** Exemplo de encaminhamento diferenciado de fluxos multimídia de dados nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao gateway multimídia no ambiente de emulação.

**(A)**

No.	Time	Source	Destination	Protocol	Length	Info
9	6.076873000	200.200.200.2	172.18.0.4	TCP	74	51295 > 6001 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SAC
10	6.077912000	172.18.0.4	200.200.200.2	TCP	74	6001 > 51295 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 M
13	6.090502000	200.200.200.2	172.18.0.4	TCP	66	51295 > 6001 [ACK] Seq=1 Ack=1 Win=29248 Len=0 TSval=
15	6.098181000	200.200.200.2	172.18.0.4	TCP	691	[TCP segment of a reassembled PDU]
16	6.101793000	200.200.200.2	172.18.0.4	TCP	66	51295 > 6001 [FIN, ACK] Seq=626 Ack=1 Win=29248 Len=0
17	6.102808000	172.18.0.4	200.200.200.2	TCP	66	6001 > 51295 [ACK] Seq=1 Ack=626 Win=30272 Len=0 TSva

▶ Frame 9: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0  
 ▶ Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:85:dc:eb (08:00:27:85:dc:eb)  
 ▼ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.4 (172.18.0.4)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0x04 (DSCP 0x01: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0000 01.. = Differentiated Services Codepoint: Unknown (0x01)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 60  
 Identification: 0xb215 (45589)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 62  
 Protocol: TCP (6)  
 ▶ Header checksum: 0x4dc1 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 172.18.0.4 (172.18.0.4)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 ▶ Transmission Control Protocol, Src Port: 51295 (51295), Dst Port: 6001 (6001), Seq: 0, Len: 0

**(B)**

No.	Time	Source	Destination	Protocol	Length	Info
298	7.853018000	200.200.200.2	172.18.0.5	TCP	74	49942 > 6002 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SAC
301	7.857940000	172.18.0.5	200.200.200.2	TCP	74	6002 > 49942 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 M
303	7.893853000	200.200.200.2	172.18.0.5	TCP	66	49942 > 6002 [ACK] Seq=1 Ack=1 Win=29248 Len=0 TSval=
304	7.893975000	200.200.200.2	172.18.0.5	TCP	691	[TCP segment of a reassembled PDU]
305	7.893992000	200.200.200.2	172.18.0.5	TCP	66	49942 > 6002 [FIN, ACK] Seq=626 Ack=1 Win=29248 Len=0
306	7.895963000	172.18.0.5	200.200.200.2	TCP	66	6002 > 49942 [ACK] Seq=1 Ack=626 Win=30272 Len=0 TSva

▶ Frame 298: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0  
 ▶ Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:c6:13:07 (08:00:27:c6:13:07)  
 ▼ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.5 (172.18.0.5)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0x08 (DSCP 0x02: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0000 10.. = Differentiated Services Codepoint: Unknown (0x02)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 60  
 Identification: 0x7abc (31420)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 62  
 Protocol: TCP (6)  
 ▶ Header checksum: 0x8515 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 172.18.0.5 (172.18.0.5)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 ▶ Transmission Control Protocol, Src Port: 49942 (49942), Dst Port: 6002 (6002), Seq: 0, Len: 0



**Figura 85.** Exemplo de recebimento de fluxos multimídia de áudio nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao gateway multimídia no ambiente de emulação.

**(A)**

No.	Time	Source	Destination	Protocol	Length	Info
346	7.884512000	200.200.200.2	188.188.188.1	UDP	106	Source port: 52915 Destination port: 7111
347	7.896361000	200.200.200.2	188.188.188.1	UDP	105	Source port: 34632 Destination port: 7112
356	7.927494000	200.200.200.2	188.188.188.1	UDP	106	Source port: 49743 Destination port: 7111
363	7.958919000	200.200.200.2	188.188.188.1	UDP	105	Source port: 51001 Destination port: 7112
415	7.975350000	200.200.200.2	188.188.188.1	UDP	106	Source port: 53525 Destination port: 7111
427	8.020241000	200.200.200.2	188.188.188.1	UDP	105	Source port: 35436 Destination port: 7112

▼ Frame 346: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 1  
 ▶ Ethernet II, Src: 08:00:27:d4:00:e9 (08:00:27:d4:00:e9), Dst: 08:00:27:2b:f2:57 (08:00:27:2b:f2:57)  
 ▼ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 188.188.188.1 (188.188.188.1)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0x44 (DSCP 0x11: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0100 01.. = Differentiated Services Codepoint: Unknown (0x11)  
 .... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 92  
 Identification: 0x022b (555)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 63  
 Protocol: UDP (17)  
 ▶ Header checksum: 0x2f99 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 188.188.188.1 (188.188.188.1)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 ▶ User Datagram Protocol, Src Port: 52915 (52915), Dst Port: 7111 (7111)

**(B)**

No.	Time	Source	Destination	Protocol	Length	Info
346	7.884512000	200.200.200.2	188.188.188.1	UDP	106	Source port: 52915 Destination port: 7111
347	7.896361000	200.200.200.2	188.188.188.1	UDP	105	Source port: 34632 Destination port: 7112
356	7.927494000	200.200.200.2	188.188.188.1	UDP	106	Source port: 49743 Destination port: 7111
363	7.958919000	200.200.200.2	188.188.188.1	UDP	105	Source port: 51001 Destination port: 7112
415	7.975350000	200.200.200.2	188.188.188.1	UDP	106	Source port: 53525 Destination port: 7111
427	8.020241000	200.200.200.2	188.188.188.1	UDP	105	Source port: 35436 Destination port: 7112

▶ Frame 347: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface 1  
 ▶ Ethernet II, Src: 08:00:27:d4:00:e9 (08:00:27:d4:00:e9), Dst: 08:00:27:2b:f2:57 (08:00:27:2b:f2:57)  
 ▼ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 188.188.188.1 (188.188.188.1)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0x48 (DSCP 0x12: Assured Forwarding 21; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0100 10.. = Differentiated Services Codepoint: Assured Forwarding 21 (0x12)  
 .... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 91  
 Identification: 0x022c (556)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 63  
 Protocol: UDP (17)  
 ▶ Header checksum: 0x2f95 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 188.188.188.1 (188.188.188.1)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 ▶ User Datagram Protocol, Src Port: 34632 (34632), Dst Port: 7112 (7112)

**Figura 86.** Exemplo de encaminhamento diferenciado de fluxos multimídia de áudio nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao gateway multimídia no ambiente de emulação.

**(A)**

No.	Time	Source	Destination	Protocol	Length	Info
319	7.933134000	200.200.200.2	172.18.0.4	UDP	106	Source port: 49743 Destination port: 7001
369	7.975990000	200.200.200.2	172.18.0.5	UDP	105	Source port: 51001 Destination port: 7002
373	7.977255000	200.200.200.2	172.18.0.4	UDP	106	Source port: 53525 Destination port: 7001
388	8.035414000	200.200.200.2	172.18.0.5	UDP	105	Source port: 35436 Destination port: 7002
390	8.035957000	200.200.200.2	172.18.0.4	UDP	106	Source port: 50809 Destination port: 7001
396	8.084572000	200.200.200.2	172.18.0.5	UDP	105	Source port: 55086 Destination port: 7002

▶ Frame 319: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface 0  
 ▶ Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:85:dc:eb (08:00:27:85:dc:eb)  
 ▶ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.4 (172.18.0.4)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0x44 (DSCP 0x11: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0100 01.. = Differentiated Services Codepoint: Unknown (0x11)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 92  
 Identification: 0x0230 (560)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 62  
 Protocol: UDP (17)  
 ▶ Header checksum: 0xfd3b [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 172.18.0.4 (172.18.0.4)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 ▶ User Datagram Protocol, Src Port: 49743 (49743), Dst Port: 7001 (7001)

**(B)**

No.	Time	Source	Destination	Protocol	Length	Info
319	7.933134000	200.200.200.2	172.18.0.4	UDP	106	Source port: 49743 Destination port: 7001
369	7.975990000	200.200.200.2	172.18.0.5	UDP	105	Source port: 51001 Destination port: 7002
373	7.977255000	200.200.200.2	172.18.0.4	UDP	106	Source port: 53525 Destination port: 7001
388	8.035414000	200.200.200.2	172.18.0.5	UDP	105	Source port: 35436 Destination port: 7002
390	8.035957000	200.200.200.2	172.18.0.4	UDP	106	Source port: 50809 Destination port: 7001
396	8.084572000	200.200.200.2	172.18.0.5	UDP	105	Source port: 55086 Destination port: 7002

▶ Frame 369: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface 0  
 ▶ Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:c6:13:07 (08:00:27:c6:13:07)  
 ▶ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.5 (172.18.0.5)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0x48 (DSCP 0x12: Assured Forwarding 21; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0100 10.. = Differentiated Services Codepoint: Assured Forwarding 21 (0x12)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 91  
 Identification: 0x0232 (562)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 62  
 Protocol: UDP (17)  
 ▶ Header checksum: 0xfd35 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 172.18.0.5 (172.18.0.5)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 ▶ User Datagram Protocol, Src Port: 51001 (51001), Dst Port: 7002 (7002)

**Figura 87.** Exemplo de recebimento de fluxos multimídia de voz nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao *gateway* multimídia no ambiente de emulação.

**(A)**

Filter: `frame.interface_id==1 and udp.port==8111 or udp.port==8112`

No.	Time	Source	Destination	Protocol	Length	Info
349	7.905572000	200.200.200.2	188.188.188.1	UDP	95	Source port: 54209 Destination port: 8111
364	7.970690000	200.200.200.2	188.188.188.1	UDP	95	Source port: 38611 Destination port: 8112
365	7.970710000	200.200.200.2	188.188.188.1	UDP	95	Source port: 53972 Destination port: 8111
424	8.007602000	200.200.200.2	188.188.188.1	UDP	95	Source port: 60430 Destination port: 8112
426	8.013005000	200.200.200.2	188.188.188.1	UDP	95	Source port: 49724 Destination port: 8111
438	8.061388000	200.200.200.2	188.188.188.1	UDP	95	Source port: 39338 Destination port: 8111

Frame 349: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface 1  
 Ethernet II, Src: 08:00:27:d4:00:e9 (08:00:27:d4:00:e9), Dst: 08:00:27:2b:f2:57 (08:00:27:2b:f2:57)  
 Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 188.188.188.1 (188.188.188.1)  
 Version: 4  
 Header length: 20 bytes  
 Differentiated Services Field: 0x84 (DSCP 0x21: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 1000 01.. = Differentiated Services Codepoint: Unknown (0x21)  
 .... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 81  
 Identification: 0x022e (558)  
 Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 63  
 Protocol: UDP (17)  
 Header checksum: 0x2f61 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 188.188.188.1 (188.188.188.1)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 User Datagram Protocol, Src Port: 54209 (54209), Dst Port: 8111 (8111)

**(B)**

Filter: `terface_id==1 and udp.port==8111 or udp.port==8112`

No.	Time	Source	Destination	Protocol	Length	Info
349	7.905572000	200.200.200.2	188.188.188.1	UDP	95	Source port: 54209 Destination port: 8111
364	7.970690000	200.200.200.2	188.188.188.1	UDP	95	Source port: 38611 Destination port: 8112
365	7.970710000	200.200.200.2	188.188.188.1	UDP	95	Source port: 53972 Destination port: 8111
424	8.007602000	200.200.200.2	188.188.188.1	UDP	95	Source port: 60430 Destination port: 8112
426	8.013005000	200.200.200.2	188.188.188.1	UDP	95	Source port: 49724 Destination port: 8111
438	8.061388000	200.200.200.2	188.188.188.1	UDP	95	Source port: 39338 Destination port: 8111

Frame 364: 95 bytes on wire (760 bits), 95 bytes captured (760 bits) on interface 1  
 Ethernet II, Src: 08:00:27:d4:00:e9 (08:00:27:d4:00:e9), Dst: 08:00:27:2b:f2:57 (08:00:27:2b:f2:57)  
 Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 188.188.188.1 (188.188.188.1)  
 Version: 4  
 Header length: 20 bytes  
 Differentiated Services Field: 0x88 (DSCP 0x22: Assured Forwarding 41; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 1000 10.. = Differentiated Services Codepoint: Assured Forwarding 41 (0x22)  
 .... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 81  
 Identification: 0x0233 (563)  
 Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 63  
 Protocol: UDP (17)  
 Header checksum: 0x2f58 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 188.188.188.1 (188.188.188.1)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 User Datagram Protocol, Src Port: 38611 (38611), Dst Port: 8112 (8112)

**Figura 88.** Exemplo de encaminhamento diferenciado de fluxos multimídia de voz nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao gateway multimídia no ambiente de emulação.

The figure consists of two screenshots, (A) and (B), showing network traffic analysis in Wireshark. Both screenshots use the same filter: `frame.interface_id==0 and udp.port==8001 or udp.pc`.

**Screenshot (A):** Shows a list of captured packets. Packet 315 is highlighted with a red dashed box and a red arrow. The packet details pane shows the following information:

- Version: 4
- Header length: 20 bytes
- Differentiated Services Field: 0x84 (DSCP 0x21: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
- 1000 01.. = Differentiated Services Codepoint: Unknown (0x21)
- .... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)
- Total Length: 81
- Identification: 0x022e (558)
- Flags: 0x02 (Don't Fragment)
- Fragment offset: 0
- Time to live: 62
- Protocol: UDP (17)
- Header checksum: 0xfd08 [validation disabled]
- Source: 200.200.200.2 (200.200.200.2)
- Destination: 172.18.0.4 (172.18.0.4)
- [Source GeoIP: Unknown]
- [Destination GeoIP: Unknown]
- User Datagram Protocol, Src Port: 54209 (54209), Dst Port: 8001 (8001)

**Screenshot (B):** Shows a list of captured packets. Packet 371 is highlighted with a red dashed box and a red arrow. The packet details pane shows the following information:

- Version: 4
- Header length: 20 bytes
- Differentiated Services Field: 0x88 (DSCP 0x22: Assured Forwarding 41; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
- 1000 10.. = Differentiated Services Codepoint: Assured Forwarding 41 (0x22)
- .... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)
- Total Length: 81
- Identification: 0x0233 (563)
- Flags: 0x02 (Don't Fragment)
- Fragment offset: 0
- Time to live: 62
- Protocol: UDP (17)
- Header checksum: 0xfcfe [validation disabled]
- Source: 200.200.200.2 (200.200.200.2)
- Destination: 172.18.0.5 (172.18.0.5)
- [Source GeoIP: Unknown]
- [Destination GeoIP: Unknown]
- User Datagram Protocol, Src Port: 38611 (38611), Dst Port: 8002 (8002)

**Figura 89.** Exemplo de recebimento de fluxos multimídia de vídeo nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao gateway multimídia no ambiente de emulação.

Filter: `frame.interface_id==1 and udp.port==9111 or udp.port==9112`

No.	Time	Source	Destination	Protocol	Length	Info
344	7.884497000	200.200.200.2	188.188.188.1	UDP	135	Source port: 53847 Destination port: 9112
345	7.884504000	200.200.200.2	188.188.188.1	UDP	135	Source port: 54793 Destination port: 9111
355	7.924662000	200.200.200.2	188.188.188.1	UDP	135	Source port: 58163 Destination port: 9111
358	7.934123000	200.200.200.2	188.188.188.1	UDP	135	Source port: 59519 Destination port: 9112
414	7.975318000	200.200.200.2	188.188.188.1	UDP	135	Source port: 38850 Destination port: 9111
419	7.989372000	200.200.200.2	188.188.188.1	UDP	135	Source port: 36375 Destination port: 9112

Frame 355: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits) on interface 1  
 Ethernet II, Src: 08:00:27:d4:00:e9 (08:00:27:d4:00:e9), Dst: 08:00:27:2b:f2:57 (08:00:27:2b:f2:57)  
 Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 188.188.188.1 (188.188.188.1)  
 Version: 4  
 Header length: 20 bytes  
 Differentiated Services Field: 0xc4 (DSCP 0x31: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 1100 01.. = Differentiated Services Codepoint: Unknown (0x31)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 121  
 Identification: 0x022f (559)  
 Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 63  
 Protocol: UDP (17)  
 Header checksum: 0x2ef8 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 188.188.188.1 (188.188.188.1)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 User Datagram Protocol, Src Port: 58163 (58163), Dst Port: 9111 (9111)

Filter: `frame.interface_id==1 and udp.port==9111 or udp.port==9112`

No.	Time	Source	Destination	Protocol	Length	Info
344	7.884497000	200.200.200.2	188.188.188.1	UDP	135	Source port: 53847 Destination port: 9112
345	7.884504000	200.200.200.2	188.188.188.1	UDP	135	Source port: 54793 Destination port: 9111
355	7.924662000	200.200.200.2	188.188.188.1	UDP	135	Source port: 58163 Destination port: 9111
358	7.934123000	200.200.200.2	188.188.188.1	UDP	135	Source port: 59519 Destination port: 9112
414	7.975318000	200.200.200.2	188.188.188.1	UDP	135	Source port: 38850 Destination port: 9111
419	7.989372000	200.200.200.2	188.188.188.1	UDP	135	Source port: 36375 Destination port: 9112

Frame 358: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits) on interface 1  
 Ethernet II, Src: 08:00:27:d4:00:e9 (08:00:27:d4:00:e9), Dst: 08:00:27:2b:f2:57 (08:00:27:2b:f2:57)  
 Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 188.188.188.1 (188.188.188.1)  
 Version: 4  
 Header length: 20 bytes  
 Differentiated Services Field: 0xc8 (DSCP 0x32: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 1100 10.. = Differentiated Services Codepoint: Unknown (0x32)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 121  
 Identification: 0x0231 (561)  
 Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 63  
 Protocol: UDP (17)  
 Header checksum: 0x2ef2 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 188.188.188.1 (188.188.188.1)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 User Datagram Protocol, Src Port: 59519 (59519), Dst Port: 9112 (9112)

**Figura 90.** Exemplo de encaminhamento diferenciado de fluxos multimídia de vídeo nos canais 1 (A) e 2 (B), ambos gerados por meio do gerador de tráfego multimídia, junto ao gateway multimídia no ambiente de emulação.

**(A)**

No.	Time	Source	Destination	Protocol	Length	Info
297	7.849496000	200.200.200.2	172.18.0.4	UDP	135	Source port: 35748 Destination port: 9001
299	7.853254000	200.200.200.2	172.18.0.5	UDP	135	Source port: 58575 Destination port: 9002
308	7.897641000	200.200.200.2	172.18.0.4	UDP	135	Source port: 54793 Destination port: 9001
309	7.897947000	200.200.200.2	172.18.0.5	UDP	135	Source port: 53847 Destination port: 9002
317	7.929686000	200.200.200.2	172.18.0.4	UDP	135	Source port: 58163 Destination port: 9001
322	7.935575000	200.200.200.2	172.18.0.5	UDP	135	Source port: 59519 Destination port: 9002

▶ Frame 297: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits) on interface 0  
 ▶ Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:85:dc:eb (08:00:27:85:dc:eb)  
 ▼ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.4 (172.18.0.4)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0xc4 (DSCP 0x31: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 1100 01.. = Differentiated Services Codepoint: Unknown (0x31)  
 .... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 121  
 Identification: 0x0226 (550)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 62  
 Protocol: UDP (17)  
 ▶ Header checksum: 0xfca8 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 172.18.0.4 (172.18.0.4)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 ▶ User Datagram Protocol, Src Port: 35748 (35748), Dst Port: 9001 (9001)

**(B)**

No.	Time	Source	Destination	Protocol	Length	Info
297	7.849496000	200.200.200.2	172.18.0.4	UDP	135	Source port: 35748 Destination port: 9001
299	7.853254000	200.200.200.2	172.18.0.5	UDP	135	Source port: 58575 Destination port: 9002
308	7.897641000	200.200.200.2	172.18.0.4	UDP	135	Source port: 54793 Destination port: 9001
309	7.897947000	200.200.200.2	172.18.0.5	UDP	135	Source port: 53847 Destination port: 9002
317	7.929686000	200.200.200.2	172.18.0.4	UDP	135	Source port: 58163 Destination port: 9001
322	7.935575000	200.200.200.2	172.18.0.5	UDP	135	Source port: 59519 Destination port: 9002

▶ Frame 299: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits) on interface 0  
 ▶ Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:c6:13:07 (08:00:27:c6:13:07)  
 ▼ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.5 (172.18.0.5)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0xc8 (DSCP 0x32: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 1100 10.. = Differentiated Services Codepoint: Unknown (0x32)  
 .... 00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 121  
 Identification: 0x0225 (549)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 62  
 Protocol: UDP (17)  
 ▶ Header checksum: 0xfca4 [validation disabled]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 172.18.0.5 (172.18.0.5)  
 [Source GeoIP: Unknown]  
 [Destination GeoIP: Unknown]  
 ▶ User Datagram Protocol, Src Port: 58575 (58575), Dst Port: 9002 (9002)

**Figura 91.** Exemplo de recebimento de fluxos multimídia de áudio, de voz, de vídeo e de dados, ambos gerados por meio do gerador de tráfego multimídia, pelos clientes multimídia H6 (canal 1, A) e H7 (canal 2, B) no ambiente de emulação.

**(A)**

No.	Time	Source	Destination	Protocol	Length	Info
3	6.076811	200.200.200.2	172.18.0.4	TCP	74	51295 > 6001 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SAC...
4	6.076842	172.18.0.4	200.200.200.2	TCP	74	6001 > 51295 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MS...
5	6.080170	200.200.200.2	172.18.0.4	UDP	95	Source port: 60353 Destination port: 8001
6	6.085768	200.200.200.2	172.18.0.4	UDP	135	Source port: 34542 Destination port: 9001
7	6.090383	200.200.200.2	172.18.0.4	TCP	66	51295 > 6001 [ACK] Seq=1 Ack=1 Win=29248 Len=0 TSval=...
8	6.093761	200.200.200.2	172.18.0.4	UDP	105	Source port: 49369 Destination port: 7001

▶ Frame 3: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)  
 ▶ Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:85:dc:eb (08:00:27:85:dc:eb)  
 ▼ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.4 (172.18.0.4)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0x04 (DSCP 0x01: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 0000 01.. = Differentiated Services Codepoint: Unknown (0x01)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 60  
 Identification: 0xb215 (45589)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 62  
 Protocol: TCP (6)  
 ▶ Header checksum: 0x4dc1 [correct]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 172.18.0.4 (172.18.0.4)  
 ▶ Transmission Control Protocol, Src Port: 51295 (51295), Dst Port: 6001 (6001), Seq: 0, Len: 0

**(B)**

No.	Time	Source	Destination	Protocol	Length	Info
3	7.826636	200.200.200.2	172.18.0.5	UDP	95	Source port: 60050 Destination port: 8002
4	7.853651	200.200.200.2	172.18.0.5	TCP	74	49942 > 6002 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SAC...
5	7.853679	172.18.0.5	200.200.200.2	TCP	74	6002 > 49942 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 M...
6	7.853809	200.200.200.2	172.18.0.5	UDP	135	Source port: 58575 Destination port: 9002
7	7.862955	200.200.200.2	172.18.0.5	UDP	105	Source port: 42067 Destination port: 7002
8	7.894653	200.200.200.2	172.18.0.5	TCP	66	49942 > 6002 [ACK] Seq=1 Ack=1 Win=29248 Len=0 TSval=...

▶ Frame 3: 95 bytes on wire (760 bits), 95 bytes captured (760 bits)  
 ▶ Ethernet II, Src: 08:00:27:af:1d:63 (08:00:27:af:1d:63), Dst: 08:00:27:c6:13:07 (08:00:27:c6:13:07)  
 ▼ Internet Protocol Version 4, Src: 200.200.200.2 (200.200.200.2), Dst: 172.18.0.5 (172.18.0.5)  
 Version: 4  
 Header length: 20 bytes  
 ▼ Differentiated Services Field: 0x88 (DSCP 0x22: Assured Forwarding 41; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
 1000 10.. = Differentiated Services Codepoint: Assured Forwarding 41 (0x22)  
 .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)  
 Total Length: 81  
 Identification: 0x0223 (547)  
 ▶ Flags: 0x02 (Don't Fragment)  
 Fragment offset: 0  
 Time to live: 62  
 Protocol: UDP (17)  
 ▶ Header checksum: 0xfd0e [correct]  
 Source: 200.200.200.2 (200.200.200.2)  
 Destination: 172.18.0.5 (172.18.0.5)  
 ▶ User Datagram Protocol, Src Port: 60050 (60050), Dst Port: 8002 (8002)