

به نام خدا

5.11

Y Cb Cr	Y	Y Cb Cr	Y
Y Cb Cr	Y	Y Cb Cr	Y

3 * 12 bit + 12 bit = 48 bit	
Y Cb Cr	Y

برای هر گروه دو پیکسلی در مجموع 48 بیت لازم داریم در نتیجه برای هر پیکسل به طور میانگین 24 بیت لازم هست پس برای هر کیفیت داریم:

$$(3840 \times 2160) \text{ pixel per frame} \times 120 \text{ frames per second} \times 24 \text{ bits per pixel} \\ = 22.24731445 \text{ Gbps}$$

$$(7680 \times 4320) \text{ pixel per frame} \times 120 \text{ frames per second} \times 24 \text{ bits per pixel} \\ = 88.98925781 \text{ Gbps}$$

$$(15360 \times 8640) \text{ pixel per frame} \times 120 \text{ frames per second} \times 24 \text{ bits per pixel} \\ = 355.9570313 \text{ Gbps}$$

5.14

(a) دلیل اصلی از بین رفتن لبه های تیز در تصویر با اتلاف پس از فشرده سازی به دلیل استفاده از الگوریتم های فشرده سازی با اتلاف است. هدف این الگوریتم ها کاهش اندازه فایل یک تصویر با حذف انتخابی برخی از داده های تصویری است که برای چشم انسان اهمیت کمتری دارند یا کمتر قابل توجه هستند. این دور ریختن انتخابی داده ها می تواند منجر به از دست رفتن اطلاعات با فرکانس بالا در تصویر شود، مانند لبه های تیز و جزئیات.

(b)

$$\text{MSE} = \frac{8}{64} * [(50 - 50)^2 + (50 - 50)^2 + (50 - 60)^2 + (50 - 80)^2 + (200 - 170)^2 \\ + (200 - 190)^2 + (200 - 200)^2 + (200 - 200)^2] = \frac{100 + 900 + 900 + 100}{8} \\ = 250 \\ \text{PSNR} = 10 \log \frac{255^2}{250} = 24.151$$

(c)

$$MSE = \frac{8}{64} * [(50 - 60)^2 + (50 - 60)^2 + (50 - 70)^2 + (50 - 90)^2 + (200 - 180)^2 + (200 - 200)^2 + (200 - 210)^2 + (200 - 210)^2]$$

$$= \frac{100 + 100 + 400 + 1600 + 400 + 100 + 100}{8} = 350$$

$$PSNR = 10 \log \frac{255^2}{350} = 22.69$$

(d) یکی از روش‌هایی که می‌تواند تأثیر چنین تغییراتی را بر اندازه‌گیری PSNR کاهش دهد، استفاده از نسخه اصلاح‌شده PSNR به نام «structural similarity index» (SSIM) است. برخلاف PSNR که فقط تفاوت‌های پیکسلی بین تصاویر اصلی و فشرده را در نظر می‌گیرد، SSIM شباهت‌های ساختاری بین دو تصویر را در نظر می‌گیرد. SSIM برای اندازه‌گیری کیفیت تصویر از سه عبارت استفاده می‌کند: luminance, contrast, structure.

$$SSIM(x, y) = (l(x, y))^{\alpha} * (c(x, y))^{\beta} * (s(x, y))^{\gamma}$$

x و y به ترتیب تصاویر اصلی و فشرده هستند.

$l(x, y)$ عبارت luminance است که شباهت روشنایی بین x و y را اندازه‌گیری می‌کند. به صورت زیر محاسبه می‌شود:

$$l(x, y) = (2 * \mu_x * \mu_y + c1) / (\mu_x^2 + \mu_y^2 + c1)$$

- μ_x و μ_y به ترتیب میانگین x و y هستند.
- $c1$ یک ثابت کوچک برای جلوگیری از تقسیم بر صفر است.

$C(x, y)$ عبارت contrast است که شباهت را در کنتراست بین x و y اندازه‌گیری می‌کند. به صورت زیر محاسبه می‌شود:

$$c(x, y) = (2 * \sigma_x * \sigma_y + c2) / (\sigma_x^2 + \sigma_y^2 + c2)$$

- σ_x و σ_y به ترتیب انحراف معیار x و y هستند.
- $c2$ یک ثابت کوچک برای جلوگیری از تقسیم بر صفر است.

$S(x, y)$ اصطلاح structure است که شباهت ساختار بین x و y را اندازه‌گیری می‌گیرد. به صورت زیر محاسبه می‌شود:

$$s(x, y) = (\sigma_{xy} + c3) / (\sigma_x * \sigma_y + c3)$$

- σ_{xy} کوواریانس بین x و y است.
- $c3$ یک ثابت کوچک برای جلوگیری از تقسیم بر صفر است.

A, β و γ عوامل وزنی هستند که اهمیت نسبی سه عبارت را کنترل می‌کنند. این مقادیر معمولاً روی 1 تنظیم می‌شوند.

$$\begin{aligned}\hat{f}_n &= trunc\left(\frac{\tilde{f}_{n-1} + \tilde{f}_{n-2}}{2}\right) \\ e_n &= f_n - \hat{f}_n \\ \tilde{e}_n &= Q[e_n] = 16 * trunc\left[\frac{255 + e_n}{16}\right] - 256 + 8 \\ \tilde{f}_n &= \hat{f}_n + \tilde{e}_n\end{aligned}$$

در $n=0$ هیچ پیش بینی نداریم در نتیجه سیگنال بازسازی شده خود سیگنال ورودی می‌شود

در $n=1$ در ابتدا از رابطه زیر استفاده می‌کنیم:

$$\hat{f}_n = trunc\left(\frac{\tilde{f}_{n-1} + \tilde{f}_{n-2}}{2}\right)$$

چون $n < 2$ در نتیجه

$$\hat{f}_1 = \tilde{f}_0$$

سپس از رابطه زیر استفاده می‌کنیم:

$$e_n = f_n - \hat{f}_n$$

و بعد از آن از رابطه زیر استفاده می‌کنیم:

$$\tilde{e}_n = Q[e_n]$$

در نهایت از رابطه زیر استفاده می‌کنیم:

$$\tilde{f}_n = \hat{f}_n + \tilde{e}_n$$

بقیه تکرار ها هم به همین روش حساب شده.

n	f_n	\hat{f}_n	e_n	\tilde{e}_n	\tilde{f}_n
0	20	-	-	-	20
1	38	20	18	24	44
2	56	32	24	24	56
3	74	50	24	24	74
4	92	65	27	24	89
5	110	81	29	24	105
6	128	97	31	24	121
7	146	113	33	40	153
8	164	137	27	24	161
9	182	157	25	24	181
10	200	171	29	24	195
11	218	188	30	24	212
12	236	203	33	40	243
13	254	227	27	24	251

(b)

در این بخش باید به جای رابطه زیر:

$$\hat{f}_n = \text{trunc}\left(\frac{\tilde{f}_{n-1} + \tilde{f}_{n-2}}{2}\right)$$

از این رابطه زیر در سمت encoder استفاده کنیم:

$$\hat{f}_n = \text{trunc}\left(\frac{f_{n-1} + f_{n-2}}{2}\right)$$

n	f_n	$\hat{f}_n(\text{encoder})$	$\hat{f}_n(\text{decoder})$	e_n	\tilde{e}_n	$\tilde{f}_n(\text{encoder})$	$\tilde{f}_n(\text{decoder})$
0	20	-	-	-	-	20	20
1	38	20	20	18	24	44	44
2	56	29	32	27	24	53	56
3	74	47	50	27	24	71	74
4	92	65	65	27	24	89	89
5	110	83	81	27	24	107	105
6	128	101	97	27	24	125	121
7	146	119	113	27	24	143	137
8	164	137	129	27	24	161	153
9	182	155	145	27	24	179	169
10	200	173	161	27	24	197	185
11	218	191	177	27	24	215	201
12	236	209	193	27	24	233	217
13	254	227	209	27	24	251	233

15.5

(a) بله، دو کلاینت محلی مختلف در یک شبکه خانگی دارای NAT می توانند به طور همزمان به یک وب سرور خارجی دسترسی داشته باشند. دستگاه NAT در شبکه به هر کلاینت یک آدرس IP محلی منحصر بفرد اختصاص می دهد و ترافیک خروجی از هر مشتری را به شماره پورت منحصر بفرد در شبکه خارجی ترسیم می کند. این به چندین کلاینت اجازه می دهد تا به طور همزمان و بدون هیچ گونه تضادی به وب سرورهای خارجی دسترسی داشته باشند.

(b) خیر، ما نمی توانیم دو سرور وب (هر دو پورت 80) را با استفاده از تنظیمات اولیه NAT در این شبکه ایجاد کنیم. دلیل آن این است که دستگاه NAT ترافیک ورودی از شبکه خارجی را به یک آدرس IP محلی و شماره پورت خاص بر اساس شماره پورت مقصد در بسته ورودی نگاشت می کند. از آنجایی که هر دو وب سرور از یک شماره پورت (یعنی پورت 80) استفاده می کنند، دستگاه NAT نمی تواند بین این دو تمایز قائل شود و نمی تواند ترافیک ورودی را به سرور صحیح ارسال کند. این مشکل را می توان با استفاده از پورت forwarding یا تنظیمات سرور مجازی در دستگاه NAT حل کرد، که اجازه می دهد ترافیک ورودی در یک پورت خاص به یک آدرس IP محلی و شماره پورت خاص هدایت شود.

(c) برای ایجاد تنها یک وب سرور (با پورت 80) در این شبکه، می توانیم از تنظیمات پورت forwarding یا سرور مجازی در دستگاه NAT استفاده کنیم تا ترافیک ورودی در پورت 80 را به آدرس IP محلی وب سرور هدایت کنیم. این به رایانه های خارجی اجازه می دهد تا با استفاده از آدرس IP عمومی دستگاه NAT به وب سرور دسترسی پیدا کنند. با این حال، اگر بخواهیم سرویس های دیگری را روی همان آدرس IP عمومی و شماره پورت میزبانی کنیم، ممکن است یک مشکل ایجاد کند. از آنجایی که ترافیک ورودی در پورت 80 به وب سرور ارسال می شود، هر سرویس دیگری که از پورت 80 استفاده می کند از شبکه خارجی غیرقابل دسترسی است. برای جلوگیری از این مشکل، می توانیم از شماره پورت دیگری برای وب سرور استفاده کنیم (به عنوان مثال، پورت 8080)، و از تنظیمات پورت forwarding یا سرور مجازی برای هدایت ترافیک ورودی در پورت 80 به آدرس IP محلی و شماره پورت سرویس دیگر استفاده کنیم. این اجازه می دهد تا هر دو سرویس از شبکه خارجی با استفاده از یک آدرس IP عمومی یکسان، اما شماره پورت های مختلف دسترسی داشته باشند.

15.10

(a) شبکه تلفن معمولی یک شبکه circuit-switched و متمرکز است که عمدتاً برای ارتباطات صوتی طراحی شده است، در حالی که اینترنت یک شبکه packet-switched و غیرمتمرکز است که از ابتدا برای پشتیبانی از انتقال داده طراحی شده است. اینترنت به دلیل انعطاف پذیری، مقیاس پذیری و مقرون به صرفه بودن امروزه به شبکه غالب تبدیل شده است.

- **متمرکز در مقابل غیرمتمرکز:** شبکه تلفن معمولی یک شبکه متمرکز است که در آن یک شرکت واحد (مانند یک ارائه دهنده مخابرات ملی) زیرساخت را کنترل می کند و نرخ تماس ها را تعیین می کند. از سوی دیگر، اینترنت یک شبکه غیرمتمرکز است که هیچ نهاد واحدی زیرساخت را کنترل نمی کند یا نرخ انتقال داده را تعیین نمی کند.
- **صدا در مقابل داده:** شبکه تلفن معمولی در درجه اول برای ارتباط صوتی طراحی شده است، با پشتیبانی محدود برای انتقال داده. از سوی دیگر، اینترنت از ابتدا برای پشتیبانی از انتقال داده ها از جمله متن، تصویر، صدا و ویدئو طراحی شده است.

(b) چالش های کلیدی برای چند رسانه ای از طریق اینترنت شامل کیفیت خدمات (QoS)، پهنای باند و ازدحام، امنیت و حریم خصوصی، و قابلیت همکاری و استانداردسازی است. تکنیک هایی مانند شکل دهی ترافیک، بافر، CDN ها، رمزگذاری و استانداردهای صنعتی را می توان برای رفع این چالش ها و اطمینان از عملکرد روان و قابل اعتماد برای برنامه های چند رسانه ای استفاده کرد.

- **کیفیت خدمات (QoS):** برنامه های چندرسانه ای، مانند پخش ویدئو و ارتباطات بلادرنگ، برای اطمینان از عملکرد روان و قابل اطمینان، به کیفیت سرویس بالا نیاز دارند. با این حال، اینترنت بهترین شبکه ای است که در آن بسته ها ممکن است با تأخیر، گم شوند یا در حین انتقال دوباره ترتیب داده شوند. برای اطمینان از QoS بالا برای برنامه های چند رسانه ای، می توان از تکنیک هایی مانند شکل دهی ترافیک، بافر و تصحیح خطا استفاده کرد.

- **پهنای باند و ازدحام:** برنامه های چند رسانه ای به پهنای باند زیادی نیاز دارند و می توانند به تراکم در شبکه کمک کنند. با مصرف بیشتر کاربران محتوای چندرسانه ای از طریق اینترنت، تقاضا برای پهنای باند و منابع شبکه افزایش می یابد که منجر به تراکم بالقوه و عملکرد کندتر می شود. برای مقابله با این چالش، می توان از تکنیک هایی مانند شبکه های تحویل محتوا (CDN)، شبکه های همتا به همتا (P2P) و جریان تطبیقی پویا استفاده کرد.
- **امنیت و حریم خصوصی:** برنامه های چند رسانه ای ممکن است شامل اطلاعات حساس یا محرمانه مانند تراکنش های مالی یا مکالمات شخصی باشد. این برنامه ها در برابر طیف وسیعی از تهدیدات امنیتی مانند استراق سمع، رهگیری داده ها و هک آسیب پذیر هستند. برای اطمینان از امنیت و حریم خصوصی برنامه های چند رسانه ای، می توان از تکنیک هایی مانند رمزگذاری، احراز هویت و کنترل دسترسی استفاده کرد.

15.15

- (a) از آنجایی که فرکانس نمونه برداری 8 کیلوهرتز است، 8000 نمونه در ثانیه وجود دارد. اگر تکه های داده هر 20 میلی ثانیه تولید شوند، در هر قطعه داده $160 = 8000 * 0.02$ نمونه وجود دارد.
- هر نمونه 8 بیت است (از آنجایی که سرعت داده 64 کیلوبیت بر ثانیه است و 8000 نمونه در ثانیه وجود دارد، هر نمونه $8 = \frac{64000}{8000}$ بیت است). بنابراین، اندازه هر قطعه $1280 = 160 * 8$ بیت یا 160 بایت است.
- (b) هنگامی که یک تکه داده در پشته پروتکل RTP/UDP/IP کپسوله می شود، header overhead وجود دارد. هدر RTP، 12 بایت، هدر UDP، 8 بایت و هدر IP معمولاً 20 بایت است. بنابراین، کل header overhead برابر با $12 + 8 + 20 = 40$ بایت است.
- (c) RTCP (Real-time Transport Control Protocol) برای نظارت و کنترل کیفیت خدمات (QoS) برای session های چند رسانه ای، از جمله تلفن اینترنتی استفاده می شود. پهنای باند اختصاص داده شده به RTCP معمولاً 5 درصد از کل پهنای باند session است. در این حالت، کل پهنای باند session 64 کیلوبیت بر ثانیه است. بنابراین، پهنای باند تخصیص یافته به $RTCP = 5\% \times 64 \text{ kbps} = 3.2 \text{ kbps}$ است.

(a) اسکایپ از TCP برای پیام‌های کنترلی استفاده می‌کند، زیرا TCP تحویل مطمئن و منظم داده‌ها را فراهم می‌کند، که برای پیام‌های کنترلی مهم است تا اطمینان حاصل شود که آنها به ترتیب صحیح و بدون از دست دادن دریافت می‌شوند. پیام‌های کنترلی در اسکایپ شامل پیام‌های سیگنال‌دهی برای راه‌اندازی و قطع تماس، و همچنین پیام‌هایی برای حفظ شبکه همپوشانی است.

(b) تمایز *super peers* و *ordinary peers* مزایای متعددی را به همراه دارد. *super peer* ها منابع بیشتری (مانند پهنای باند و قدرت پردازش) نسبت به *ordinary peers* دارند، بنابراین می‌توانند به عنوان *relays* برای تماس‌های بیشتر عمل کنند و ترافیک بیشتری را مدیریت کنند. این به کاهش بار روی *ordinary peers* و بهبود عملکرد کلی سیستم کمک می‌کند. *super peers* همچنین اتصالات پایدارتری دارند (مانند اتصالات اینترنت پرسرعت) که باعث می‌شود به عنوان *relays* قابل اعتمادتر باشند.

(c) در یک کنفرانس N کاربر، هر کاربر باید کپی خود را از جریان صوتی برای $N - 1$ کاربران دیگر ارسال کند، بنابراین $N(N - 1)$ نسخه از جریان صوتی در کل تحویل داده می‌شود.

(d) با درخواست از هر کاربر برای ارسال جریان خود به شروع‌کننده کنفرانس، تعداد جریان‌هایی که باید به کل کنفرانس ارسال شود به $N - 1$ کاهش می‌یابد، زیرا شروع‌کننده فقط باید یک جریان ترکیبی را برای هر کاربر ارسال کند. مزیت اصلی این راه حل این است که استفاده از پهنای باند کلی را کاهش می‌دهد، زیرا هر کاربر تنها نیاز به ارسال یک جریان به جای جریان‌های $N - 1$ دارد. با این حال، این راه حل همچنین یک نقطه شکست را معرفی می‌کند (آغاز‌کننده کنفرانس)، و اگر شروع‌کننده دور از برخی از شرکت‌کنندگان کنفرانس قرار داشته باشد، می‌تواند منجر به تأخیر بیشتر شود. برای بهبود این راه حل، می‌توان از شروع‌کننده‌های متعدد برای توزیع بار و کاهش خطر یک نقطه شکست استفاده کرد.

(e) انتقال سرویس Skype به پلتفرم Microsoft Azure Cloud دارای چندین مزیت است، از جمله بهبود مقیاس پذیری، قابلیت اطمینان و امنیت. پلتفرم‌های ابری می‌توانند منابع منعطف و قدرتمندی را فراهم کنند که به راحتی می‌توان آن‌ها را در صورت نیاز کوچک یا بزرگ کرد، که می‌تواند به کنترل نوسانات در ترافیک و بهبود عملکرد کلی کمک کند. پلتفرم‌های ابری همچنین می‌توانند ویژگی‌های امنیتی قوی و حفاظت از داده‌ها را ارائه دهند که برای سرویس‌های ارتباطی مانند اسکایپ مهم است. با این حال، استفاده از پلتفرم ابری دارای معایبی است، از جمله افزایش هزینه‌ها، نگرانی‌های حفظ حریم خصوصی داده‌ها و قفل شدن قابلیت انتخاب. علاوه بر این، مهاجرت به یک پلتفرم جدید می‌تواند فرآیندی پیچیده و زمان‌بر باشد که نیازمند برنامه‌ریزی و اجرای دقیق برای اطمینان از انتقال روان با حداقل اختلال برای کاربران است.

DASH (Dynamic Adaptive Streaming over HTTP) یک پروتکل استریم است که از HTTP برای ارائه محتوای ویدیویی و صوتی از طریق اینترنت استفاده می کند. DASH مشکلات مربوط به saw-tooth در TCP را با استفاده از جریان بیت تطبیقی که به مشتری اجازه می دهد تا بخش هایی از رسانه را با نرخ بیت های مختلف بسته به پهنای باند موجود درخواست کند، برطرف می کند. این بدان معنی است که DASH می تواند نرخ بیت را به صورت پویا بر اساس شرایط فعلی شبکه تنظیم کند تا از مسدود شدن مداوم صف داده در هنگام تراکم یا خطاهای کانال جلوگیری کند.

علاوه بر پخش نرخ بیت تطبیقی، DASH پشتیبانی های دیگری را نیز برای پخش ارائه می دهد که در HTTP اصلی وجود ندارند. که شامل:

1. **تقسیم بندی رسانه:** DASH محتوای رسانه را به بخش های کوچک و قابل مدیریتی تقسیم می کند که می توانند از طریق HTTP تحویل داده شوند و امکان تحویل کارآمدتر و کنترل بهتر بر بافر را فراهم می کند.

2. **Media presentation description (MPD):** DASH از یک فایل manifest مبتنی بر XML به نام MPD استفاده می کند که اطلاعاتی درباره بخش های رسانه موجود، نرخ بیت آنها و سایر ابردادها ارائه می دهد. این به مشتری اجازه می دهد تا بخش های مناسب را بر اساس شرایط فعلی شبکه انتخاب کند.

3. **سوئیچینگ کنترل شده توسط سرور:** DASH به سرور اجازه می دهد تا رفتار سوئیچینگ نرخ بیت را بر اساس شرایط فعلی شبکه و قابلیت های کلاینت کنترل کند. این یک تجربه پخش روان را برای کاربر تضمین می کند، حتی در حضور شرایط مختلف شبکه.

4. **رمزگذاری:** DASH از رمزگذاری محتوای رسانه با استفاده از پروتکل های رمزگذاری استاندارد مانند AES-128 برای اطمینان از امنیت و حریم خصوصی محتوا پشتیبانی می کند.

به طور کلی، DASH با رسیدگی به مشکلات مربوط به رفتار saw-tooth در TCP و ارائه پشتیبانی های اضافی برای تحویل کارآمد و کنترل بر محتوای رسانه، راه حل قوی تر و انعطاف پذیرتری برای پخش رسانه از طریق HTTP ارائه می دهد.

دو برنامه چند رسانه ای اینترنتی محبوب که از DASH و WebRTC استفاده می کنند به ترتیب Netflix و Google Meet هستند.

Netflix یک سرویس پخش ویدیوی محبوب است که از DASH برای قابلیت های پخش نرخ بیت تطبیقی خود استفاده می کند. DASH به Netflix اجازه می دهد تا با تنظیم پویا نرخ بیت بر اساس پهنای باند موجود، محتوای ویدیویی با کیفیت بالا را به کاربران با شرایط شبکه متفاوت ارائه دهد. این امر یک تجربه پخش روان را برای کاربران تضمین می کند، حتی در صورت وجود شرایط شبکه نوسان. در مورد استفاده Netflix از DASH از طریق وبلاگ فنی آنها و نشریات صنعتی مختلف که در مورد پروتکل های جریان بحث می کنند مطلع شدم.

Google Meet یک برنامه ویدئو کنفرانس است که از WebRTC برای قابلیت های ارتباطی بلادرنگ خود استفاده می کند. WebRTC یک فناوری مبتنی بر مرورگر است که ارتباط صوتی و تصویری بلادرنگ بین مرورگرها را بدون نیاز به پلاگین یا نرم افزار اضافی امکان پذیر می کند. Google Meet از WebRTC برای تسهیل تماس های ویدیویی و صوتی با کیفیت بالا بین کاربران، صرف نظر از مکان یا دستگاه آنها، استفاده می کند. من در مورد استفاده Google Meet از WebRTC از طریق وب سایت آنها و انتشارات مختلف صنعتی که در مورد فن آوری های ارتباطی بلادرنگ بحث می کنند مطلع شدم.

این برنامه ها برای استفاده خاص خود انتخاب شده اند، زیرا به ارائه مطمئن و کارآمد محتوای چندرسانه ای از طریق اینترنت نیاز دارند. DASH و WebRTC قابلیت های لازم را برای ارائه محتوای ویدیویی و صوتی با کیفیت بالا در زمان واقعی ارائه می دهند، در حالی که با شرایط مختلف شبکه سازگار می شوند و تجربه کاربری یکپارچه را ارائه می دهند.