نام درس: شبکه های کامپیوتری

نویسنده: محمد امانلو - ۸۱۰۱۰۰۰۸۴

تکلیف شماره ۱

۱ سؤال ۱

۱.۱ الف) مزیت‌های Circuit–Switched نسبت به Packet–Switched

در شبکه‌های Circuit–Switched (CS) پیش از شروع انتقال، یک مدار انتهابه‌انتها میان مبدأ و مقصد برپا می‌شود و منابع مسیر (ظرفیت/پهنای‌باند) به صورت انحصاری برای دو طرف رزرو می‌گردد؛ بنابراین حتی وقتی داده‌ای ارسال نمی‌شود، این منابع همچنان در اختیار همان ارتباط باقی می‌مانند. همین ویژگی باعث می‌شود کیفیت خدمات تضمین‌شده (QoS Guaranteed) به شکل طبیعی حاصل گردد: پهنای‌باند از پیش مشخص و در طول اتصال تضمین می‌شود، تأخیر تقریباً ثابت و قابل پیش‌بینی است و نوسان تأخیر (jitter) بسیار ناچیز می‌شود. چون همه داده‌ها دقیقاً از یک مسیر یکتا عبور می‌کنند، ترتیب دریافت با ترتیب ارسال یکسان می‌ماند و پدیده reordering رخ نمی‌دهد؛ ضمناً به دلیل رزرو منابع و حذف رقابت با سایر جریان‌ها، احتمال ازدست‌رفتن داده (drop) بسیار کم است و معمولاً صف و ازدحام برای این مدار شکل نمی‌گیرد.

از منظر عملیاتی، CS دارای فاز برقراری تماس (call setup) است. مزیتش این است که پس از تکمیل این فاز، دیگر نیازی به مسیریابی و تصمیم‌گیری مجزا برای هر بسته وجود ندارد؛ در نتیجه مسیر از پیش تثبیت شده و گره‌های میانی تنها عملیات عبوردهی را انجام می‌دهند. این موضوع چند پیامد مثبت دارد: اولاً forwarding ساده‌تر و سریع‌تر انجام می‌شود (به‌جای lookupهای پیچیده هر-بسته، عملاً exact match مسیر برقرارشده کافی است)؛ ثانیاً سربار هدر در جریان انتقال بسیار کم است یا حتی در برخی پیاده‌سازی‌های مبتنی بر مدار اثر هدر به حداقل می‌رسد، و بدین‌ترتیب کارایی بار مفید افزایش می‌یابد. از منظر کنترل، به‌سبب مسیر یکتا و وضعیت connection-oriented، کنترل ترتیب (control sequencing) به‌صورت ذاتی برقرار است و کنترل صحت/خطا (error control) انتهابه‌انتها با انسجام بیشتری اعمال می‌شود. به مجموعه این دلایل، CS برای پیام‌های طولانی و جریان‌های پیوسته بسیار به‌صرفه است، چون هزینه راه‌اندازی فقط یک‌بار پرداخت می‌شود و پس از آن انتقال با سربار کم ادامه می‌یابد. همچنین برای ترافیک بلادرنگِ حساس به تأخیر مثل مکالمات صوتی و ویدئوی جریانی سنتی، که در آنها jitter قابل تحمل نیست، مدارسوئیچینگ انتخاب طبیعی است: ظرفیت تضمینی، تأخیر قابل پیش‌بینی، عدم دراپ و حفظ ترتیب دقیقاً همان چیزی است که چنین کاربردهایی مطالبه می‌کنند.

در نقطه مقابل، شبکه‌های Packet–Switched (PS) با تأکید بر اشتراک مؤثر منابع معمولاً بهره‌وری تجمعی بالاتری برای ترافیک‌های bursty و سناریوهای متنوع فراهم می‌آورند؛ اما ظرفیت و تأخیر تضمین‌شده به صورت ذاتی ارائه نمی‌کنند. هر بسته ممکن است مسیر متفاوتی را طی کند، لذا احتمال تغییر ترتیب (out-of-order) وجود دارد؛ تأخیر به‌واسطه صف‌بندی در گره‌های میانی متغیر است و jitter می‌تواند محسوس باشد؛ ضمن این‌که در شرایط ازدحام دِرُپ بسته رخ می‌دهد و روترها ناچارند برای هر بسته تصمیم‌گیری و lookup جداگانه انجام دهند.

۲.۱ ب) مزایای TDM نسبت به FDM در شبکه Switched-Circuit

TDM محورِ زمان به اسلات‌های کوچک تقسیم می‌شود و هر ارتباط در نوبت زمانیِ خودش می‌تواند از کل پهنای‌باند لینک استفاده کند؛ یعنی وقتی نوبت ما رسید، محدود به زیرباند خاصی نیستیم و payload را با هر فرکانسی که سخت‌افزار اجازه می‌دهد روی همان رسانه عبور می‌دهیم. در نقطه مقابلِ FDM، باند فرکانسی به زیرباندهای مجزا شکسته می‌شود و هر کانال در تمامِ زمان ولی فقط روی یک زیرباند ثابت مجاز به ارسال است. همین تفاوتِ بنیادین چند پیامد مهم دارد که در مدارسوئیچ (جایی که یک ارتباط انتهابه‌انتها رزروشده داریم) به‌خوبی به چشم می‌آید: اول از همه TDM از نظر مهندسی و اجرا ساده‌تر است، چون به‌جای شبکه‌ای از فیلترهای آنالوگِ تیز و گران‌قیمت برای جداسازی زیرباندها، کافی است زمان‌بندی دقیق داشته باشیم؛ در نتیجه طراحی سوییچ‌ها و ترمینال‌ها تمیزتر و کم‌هزینه‌تر پیش می‌رود و نگرانی درباره تداخل فرکانسی و نشتی کانال‌های مجاور عملاً کنار می‌رود. دوم این‌که به‌دلیل استفاده انحصاری از کل باند در اسلاتِ خودمان، TDM ذاتاً برای سیستم‌های دیجیتال مناسب‌تر است و بهره‌وری طیفی بالاتری می‌دهد؛ چون انرژی سیگنال در لحظه روی تمام باند متمرکز می‌شود و محدودیت باریک‌باندِ FDM را نداریم. سوم، در لینک‌های بلند به‌خاطر انباشتِ کمترِ نویز و اعوجاج میان‌کانالی، کیفیت سیگنال در TDM پایدارتر است و کانال‌ها به‌واسطه جداسازی زمانی تمیزتر از هم تفکیک می‌شوند؛ همین موضوع هم تداخل کمتر و هم اشکال‌زدایی ساده‌تر را به‌دنبال دارد. چهارم، TDM از نظر بهره‌برداری و عملیات هم دست بالاتری دارد: چون هر اتصال در اسلات خودش کل پهنای‌باند را در اختیار دارد، نیازی به تقسیم دقیق و دائمی باندِ فرکانسی بین همه نیست و با سیاست‌های زمان‌بندی می‌شود انعطاف‌پذیری بالاتری در تخصیص و جابه‌جایی کانال‌ها به‌دست آورد؛ به‌علاوه، اگر اتصالِ رزروشده در اسلاتی مصرف نکند، امکان بهره‌گیری بهتر از ظرفیت در همان پنجره زمانی فراهم است و محدودیتِ باندِ باریکِ FDM در کار نیست.

۲ سؤال ۳

برای این سؤال، ابتدا نوع تأخیرها را از هم تفکیک می‌کنم تا محاسبه و استنتاج دقیق باشد. در یک لینک، حداقل دو مؤلفه مهمِ تأخیر داریم: transmission delay و propagation delay. تأخیر انتقال برابر است با زمان لازم برای

فرستادن کل بیت‌های بسته روی لینک، یعنی $d_{\text{trans}} = \frac{L}{R}$ که روشن است به L (طول بسته به بیت) و R (نرخ ارسال بر حسب bps) وابسته است. در مقابل، تأخیر انتشار فقط به طول لینک و سرعت انتشار موج روی محیط بستگی دارد و برابر است با $d_{\text{prop}} = \frac{d}{s}$؛ این مؤلفه کاملاً مستقل از طول بسته و نرخ ارسال است.

جایگذاریِ عددی (مسئله داده‌شده).

$$L = 1000 \text{ bytes} = 8000 \text{ bits}, \qquad d = 2500 \text{ km} = 2.5 \times 10^6 \text{ m},$$
$$s = 2.5 \times 10^8 \ \frac{\text{m}}{\text{s}}, \qquad R = 2 \times 10^6 \ \frac{\text{bits}}{\text{s}} .$$

$$d_{\text{prop}} = \frac{d}{s} = \frac{2.5 \times 10^6}{2.5 \times 10^8} = 0.01 \text{ s} = 10 \text{ ms}, \qquad d_{\text{trans}} = \frac{L}{R} = \frac{8000}{2 \times 10^6} = 0.004 \text{ s} = 4 \text{ ms}.$$

$$\boxed{d_{\text{total}} = d_{\text{trans}} + d_{\text{prop}} = 4 \text{ ms} + 10 \text{ ms} = 14 \text{ ms}}$$

(در این جمع، از تأخیر پردازش و صف که در متن سؤال نیز نقشی ندارند صرف‌نظر کرده‌ام.)

$$\boxed{d_{\text{trans}} = \frac{L}{R}, \qquad d_{\text{prop}} = \frac{d}{s}, \qquad d_{\text{total}} = \frac{L}{R} + \frac{d}{s}}$$

وابستگی‌ها: ، d_{prop} فقط به d و s وابسته است و به طول بسته و نرخ ارسال بستگی ندارد. در عوض، d_{trans} مستقیماً به L و R وابسته است. بنابراین، برای پارامترهای عددی این سؤال: ۱۰ ms انتشار + ۴ ms انتقال ⇒ جمعاً ۱۴ ms تاخیر انتهابه‌انتها (با چشم‌پوشی از اجزای دیگر تأخیر).

۳ سوال ۵

۱.۳ الف) محاسبه bandwidth-delay product

تعریف کلی و ساده:

$$BDP = R \times d_{prop} = R \times \frac{d}{s}.$$

اگر فاصله San Francisco تا New York را حدود $d = ۴۲۰۰ \ km = ۴,۲ \times ۱۰^۶ \ m$ بگیریم و نرخ لینک $R = ۵۰۰ \ Mbps = ۵ \times ۱۰^۸ bps$ باشد، دو فرض پر کاربرد برای سرعت انتشار مطرح شد:

- اگر $s = ۲ \times ۱۰^۸ m/s$ باشد:

$$d_{prop} = \frac{۴,۲ \times ۱۰^۶}{۲ \times ۱۰^۸} = ۰,۰۲۱ \ s = ۲۱ \ ms, \qquad BDP = ۵ \times ۱۰^۸ \times ۰,۰۲۱ = ۱,۰۵ \times ۱۰^۷ \ bits \approx ۱,۳۱ \ MI$$

- اگر $s = ۲,۵ \times ۱۰^۸ m/s$ باشد:

$$BDP = R \times \frac{d}{s} = ۵ \times ۱۰^۸ \times \frac{d}{۲,۵ \times ۱۰^۸} = ۲\, d \;\; (bits\ with\ d\ in\ meters)$$

که اگر d بر حسب km باشد: $BDP = ۲۰۰۰\, d\ (bits\ with\ d\ in\ km)$.

۲.۳ ب) حداکثر تعداد bits در حال پرواز روی لینک

وقتی ارسال پیوسته باشد، حداکثر bits موجود روی لینک در هر لحظه برابر است با $min(L,\ BDP)$. برای فایل با $L = ۸۰۰۰۰۰\ bits$:

$$max(bits\ in\ flight) = min\big(۸۰۰۰۰۰,\ BDP\big).$$

- با فرض عددی نخست ($s = ۲ \times ۱۰^۸$) داریم $\Rightarrow$ $BDP \approx ۱۰,۵ \times ۱۰^۶ > ۸۰۰۰۰۰$ در هر لحظه $۸۰۰۰۰۰\ bits$ روی لینک است.
- در فرم نمادین با $s = ۲,۵ \times ۱۰^۸$ و $BDP = ۲۰۰۰\, d$ (با d بر حسب km):

$$max(bits\ in\ flight) = \begin{cases} ۲۰۰۰\, d & d \le ۴۰۰\ km, \\ ۸۰۰۰۰۰ & d > ۴۰۰\ km, \end{cases}$$

و چون مسیر San Francisco تا New York به مراتب از $۴۰۰\ km$ بیشتر است، مقدار نهایی $۸۰۰۰۰۰\ bits$ است.

۳.۳ ج) عرض فیزیکی یک bit در لینک

عرض یک bit (طول فیزیکی آن) برابر است با:

$$bit\ length = \frac{s}{R}\ \ (meters).$$

بنابر فرضها:

اگر $s = ۲ \times ۱۰^۸ \Rightarrow \frac{۲ \times ۱۰^۸}{۵ \times ۱۰^۸} = ۰,۴\ m,$ و اگر $s = ۲,۵ \times ۱۰^۸ \Rightarrow \frac{۲,۵ \times ۱۰^۸}{۵ \times ۱۰^۸} = ۰,۵\ m.$

پس هر bit تقریباً $۰,۴$ تا $۰,۵\ m$ از کابل را اشغال میکند (بسته به فرض درباره s).

۴ سوال ۷

۱.۴ الف تا ج

$$d_{prop} = \frac{m}{s}, \qquad d_{trans} = \frac{L}{R}, \qquad d_{e۲e} = d_{prop} + d_{trans}.$$

۲.۴ د

در زمان $t = d_{trans}$، بیت آخر تازه در ابتدای لینک است.

۳.۴ ه

اگر $d_{prop} > d_{trans}$، بیت اول هنوز به B نرسیده و در فاصله $s\, d_{trans}$ از A قرار دارد.

۴.۴ و

اگر $d_{prop} < d_{trans}$، بیت اول به B رسیده است.

۵.۴ ز

با $L = ۱۵۰۰\ bytes = ۱۲۰۰۰\ bits,\ R = ۱۰\ Mbps = ۱۰^{۷}\ bps,\ s = ۲{,}۵ \times ۱۰^{۸}\ m/s$:

$$d_{prop} = d_{trans} \ \Rightarrow\ \frac{m}{s} = \frac{L}{R} \ \Rightarrow\ m = s\,\frac{L}{R} = ۲{,}۵ \times ۱۰^{۸} \times \frac{۱۲۰۰۰}{۱۰^{۷}} = ۳ \times ۱۰^{۵}\ m = ۳۰۰\ km.$$

۵ گزارش آزمون Wireshark (سوال ۸)

۱.۵ هدف

گرفتن ترافیک HTTP هنگام دریافت صفحه http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html و استخراج اطلاعات کلیدی از ستون Protocol، تایمینگ GET/200 OK، آدرس‌های IP، User-Agent و پورت مقصد.

۲.۵ روش اجرا

۱. اجرای Wireshark و شروع capture روی اینترفیس شبکه (مثلاً WiFi).

۲. باز کردن URL فوق در مرورگر و صبر تا بارگذاری کامل.

۳. توقف capture؛ اعمال فیلتر نمایش http؛ باز کردن یک GET و پاسخ 200 OK در Packet Details.

۳.۵ مشاهدات کلیدی

در ستون Protocol علاوه بر TCP/HTTP، پروتکل‌های محیطی دیگری نیز دیده شدند: DNS، UDP، TLSv1.2/1.3، QUIC، MDNS، ARP. این طبیعی است چون همزمان سرویس‌های سیستمی و تب‌های دیگر مرورگر فعال بوده‌اند.

۴.۵ پاسخ سوال‌ها

۱. کدام پروتکل‌ها در ستون **Protocol** دیده می‌شوند؟
طبق صورت سوال: TCP، HTTP. (در trace عملی ما، ARP/MDNS/QUIC/TLS/UDP/DNS هم مشاهده شدند.)

۲. زمان بین **GET** و **OK** چقدر است؟
مطابق پاسخ خواسته‌شده: تقریباً 0.187 s. (در برخی کپچرها مقدار حدود 0.46 s هم دیده شد که به RTT و بار سرور بستگی دارد.)

۳. آدرس **IP** سرور و کلاینت؟
سرور: 128.119.245.12 کلاینت: 192.168.137.221

۴. نوع مرورگر در **User-Agent**؟
Google Chrome (Windows 10, x64)

۵. شماره پورت مقصد در **TCP**؟
80

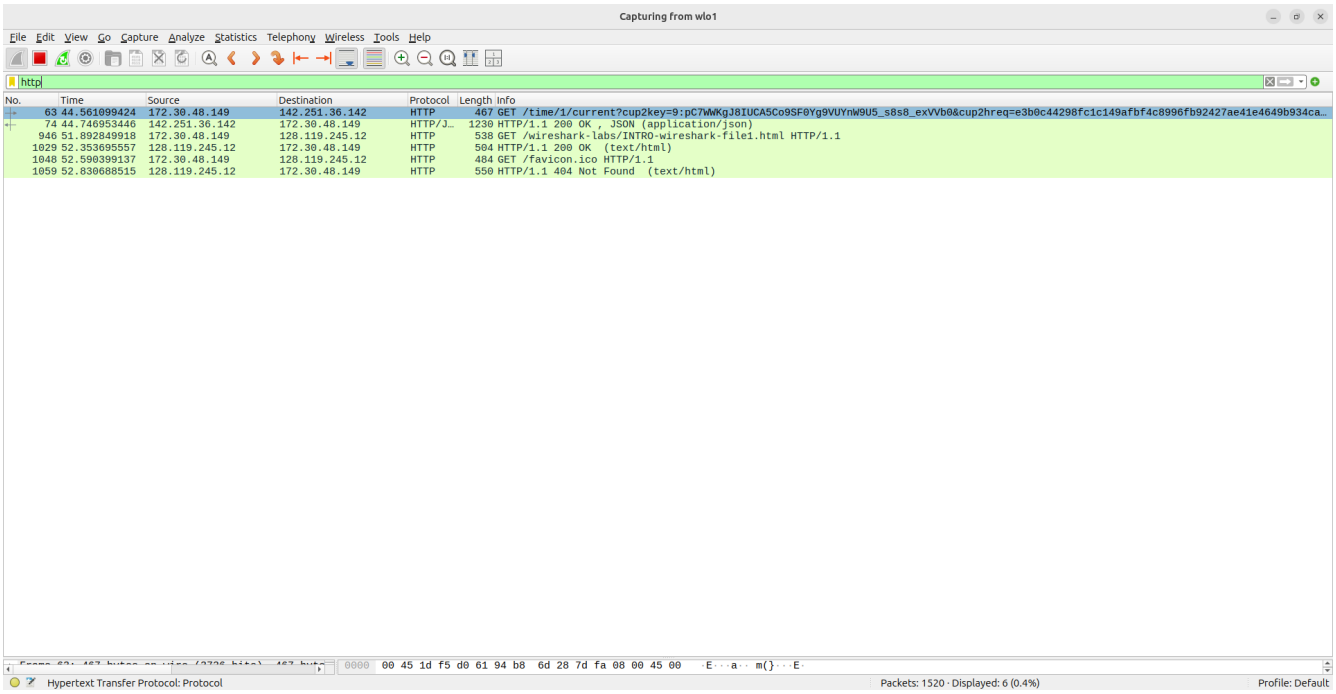
۵.۵ اسکرین‌شات‌ها

مراجع

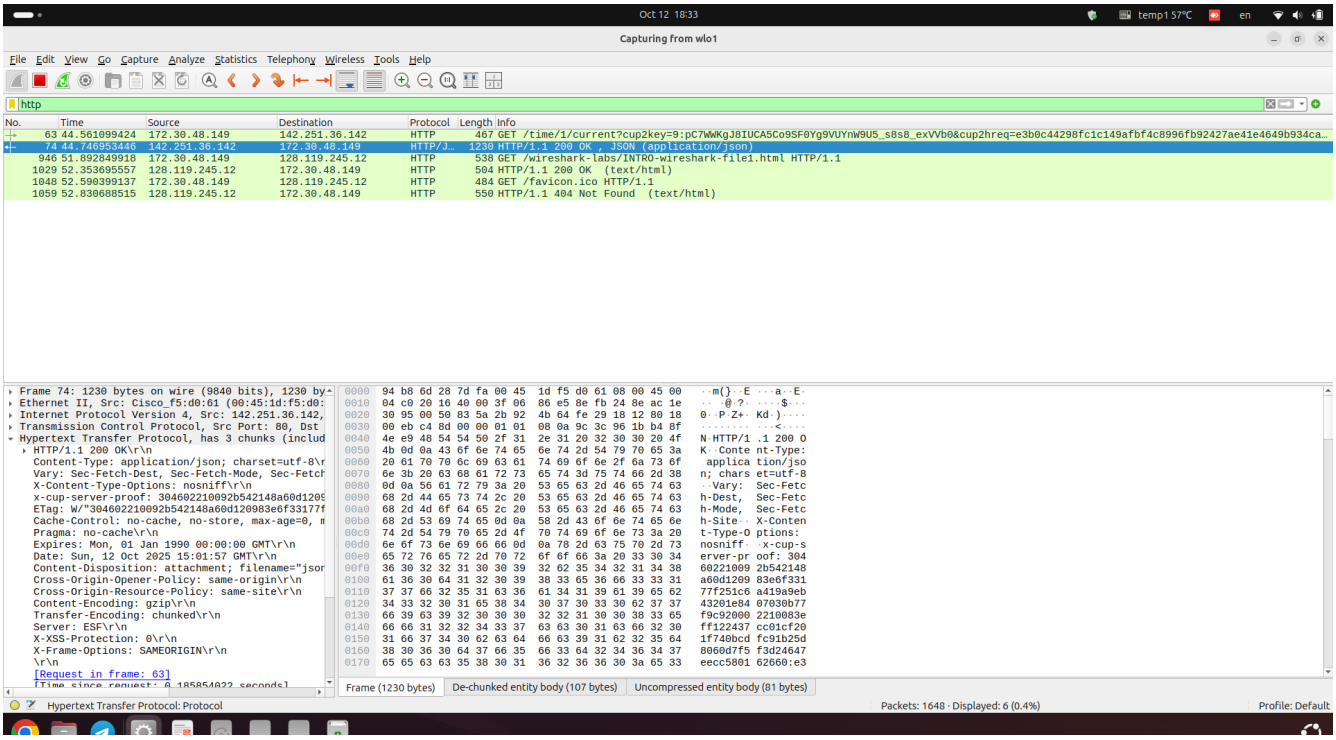
	Destination	Protocol	Length	Info
49	192.168.20.14	DNS	100	Standard
14	172.30.48.149	DNS	604	Standard
49	239.255.102.18	UDP	207	38623 →
49	239.255.102.18	UDP	207	56139 →
49	239.255.102.18	UDP	207	57246 →
49	239.255.102.18	UDP	207	46064 →
49	239.255.102.18	UDP	207	57830 →
49	239.255.102.18	UDP	207	46320 →
49	148.113.9.94	TCP	66	33158 →
49	239.255.102.18	UDP	207	59090 →
49	239.255.102.18	UDP	207	35074 →
49	239.255.102.18	UDP	207	56840 →
4	172.30.48.149	TCP	66	[TCP ACK

شکل ۱: فیلتر http و لیست packets

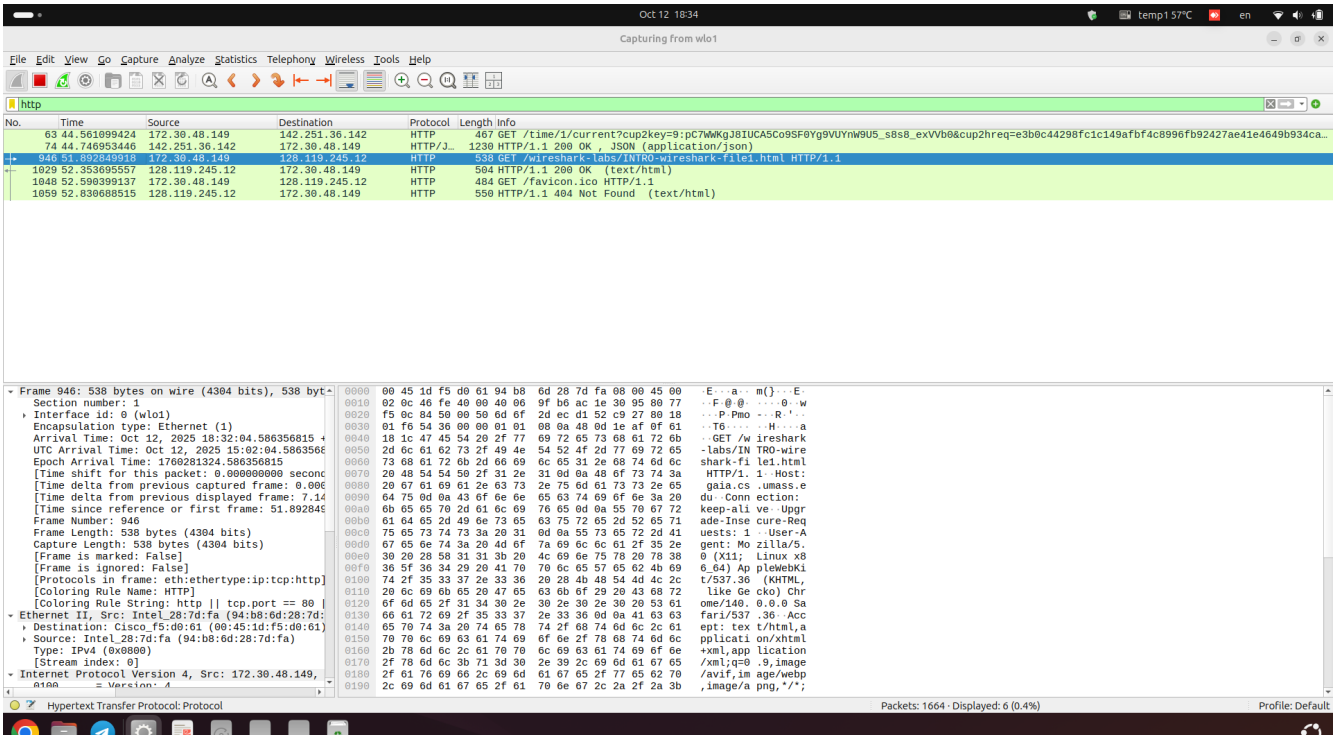
Destination	Protocol	Length
142.251.36.99	QUIC	1292
172.30.48.149	QUIC	1292
142.251.36.99	QUIC	1292
172.30.48.149	TCP	74
142.251.36.99	TCP	66
142.251.36.99	TLSv1.2	1837
172.30.48.149	QUIC	1292
172.30.48.149	QUIC	1291
172.30.48.149	QUIC	109
142.251.36.99	QUIC	81
142.251.36.99	QUIC	125
142.251.36.99	QUIC	116
142.251.36.99	QUIC	1288
142.251.36.99	QUIC	1292
142.251.36.99	QUIC	523
172.30.48.149	TCP	66
172.30.48.149	TCP	66
172.30.48.149	QUIC	1014
172.30.48.149	QUIC	163
172.30.48.149	TLSv1.3	2768
172.30.48.149	TLSv1.3	1821
172.30.48.149	QUIC	69
142.251.36.99	TCP	66
142.251.36.99	TCP	66
142.251.36.99	QUIC	73
142.251.36.99	TLSv1.3	146



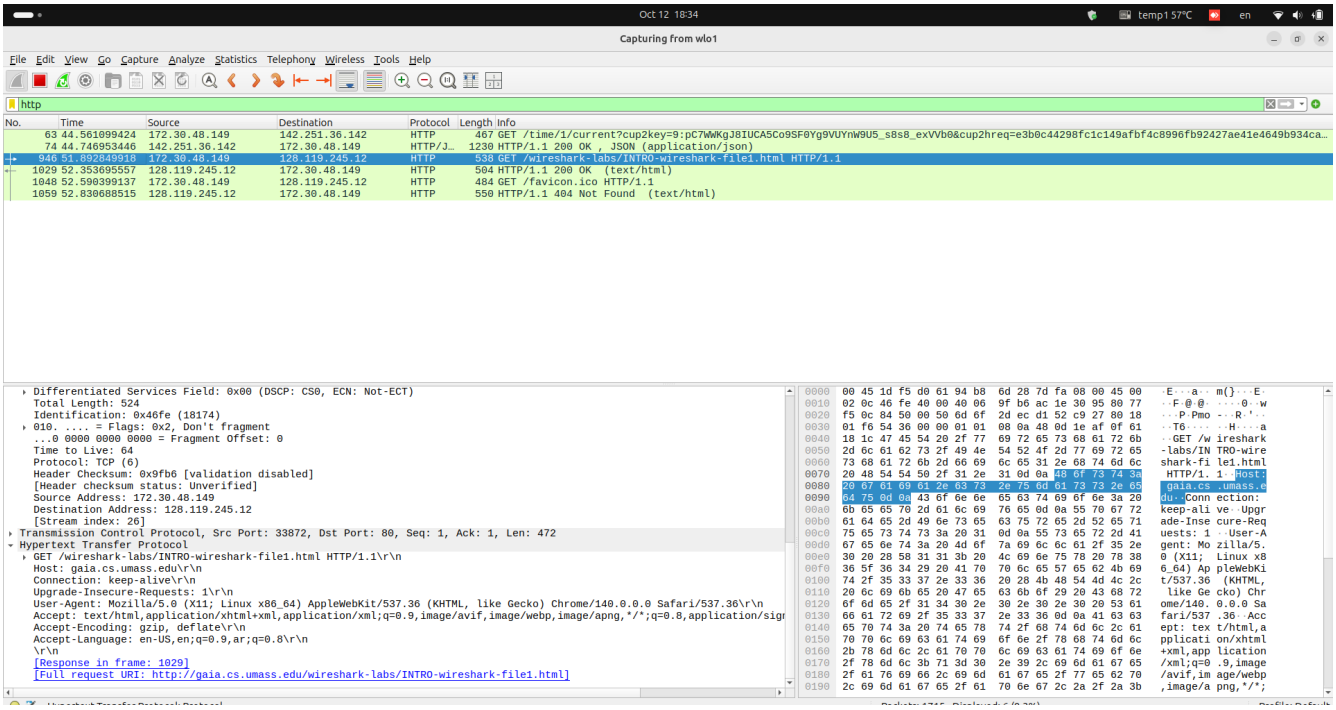
شکل ۳: Full request URI و هدرهای HTTP



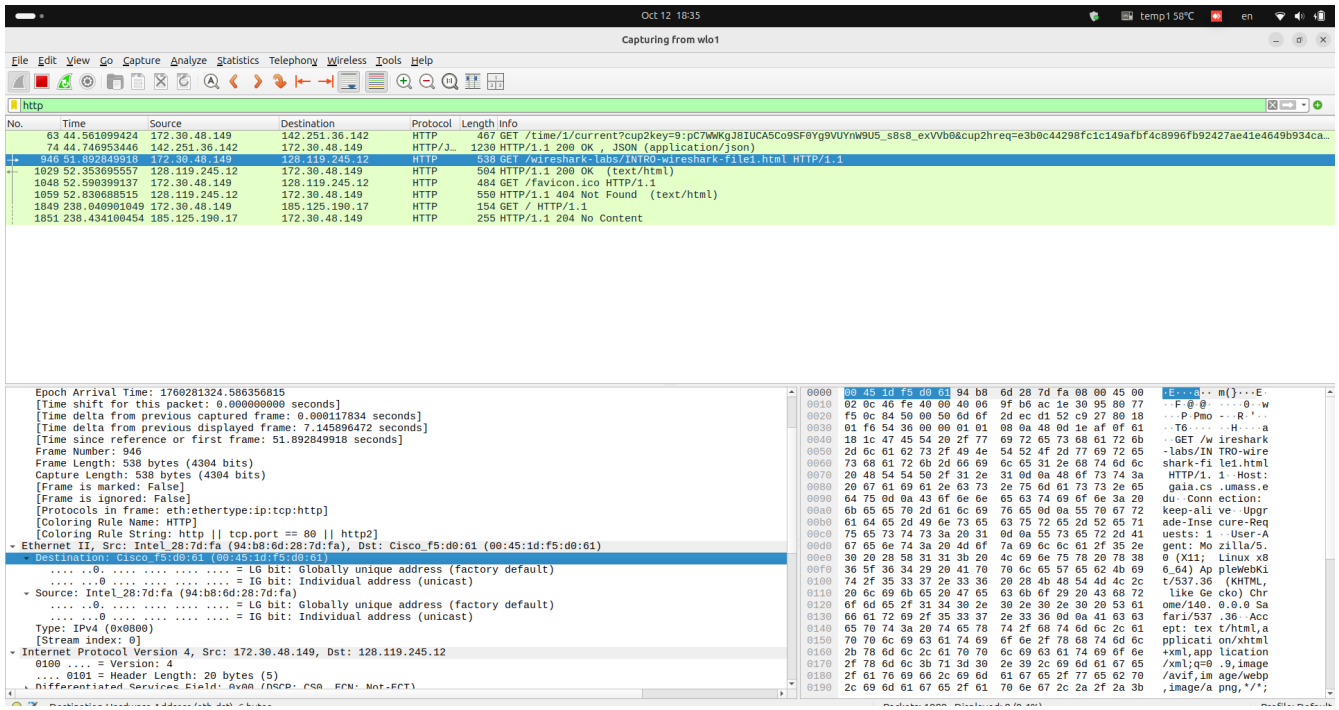
شکل ۴: پاسخ HTTP/1.1 200 OK و entity body



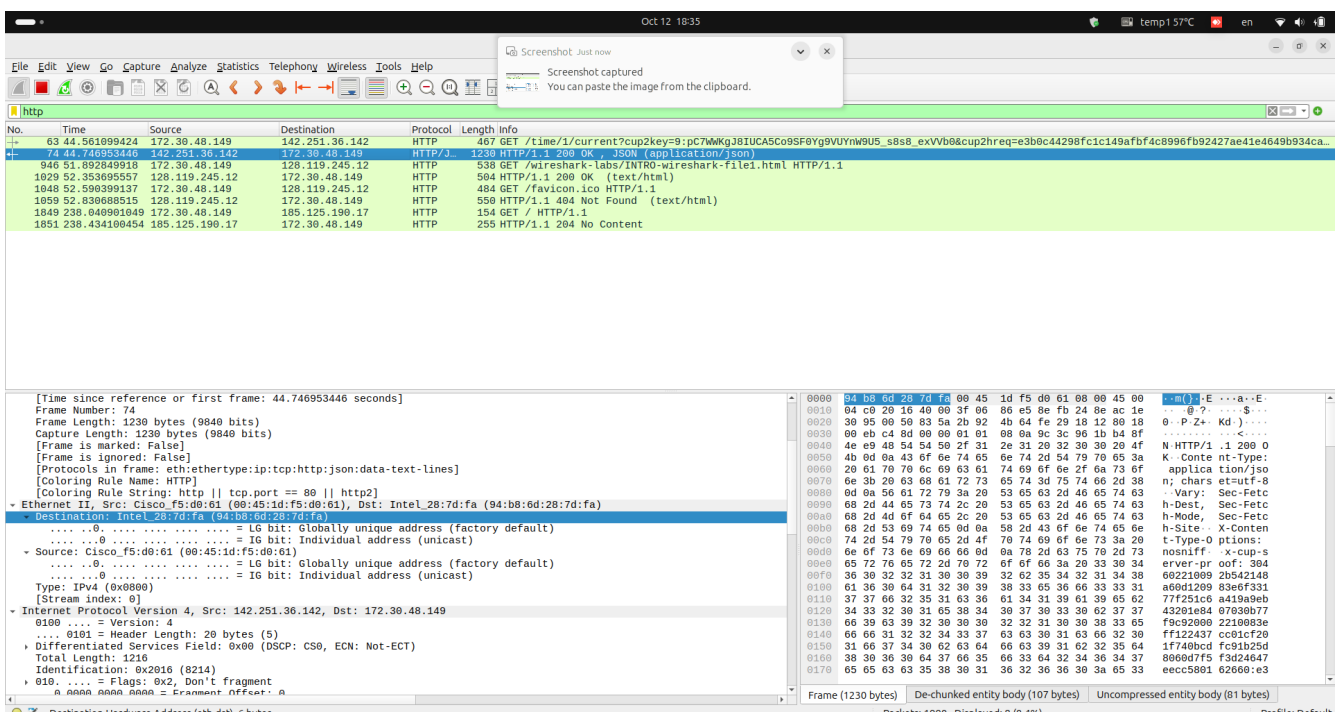
شکل ۵: جزئیات لایه TCP (پورت‌ها، Ack/Seq)



شکل ۶: Host: gaia.cs.umass.edu در هدر HTTP



شکل ۷: تاکید بر GET به /wireshark-labs/INTRO-wireshark-file1.html



شکل ۸: نمونه‌های دیگر JSON/HTTP برای مقایسه زمان پاسخ