



تمرین شماره پنج + حل تمرین

درس شبکه های کامپیوتری

مدرس: دکتر هاشمی

\*\*\*

۱- متن زیر یک پیام HTTP GET است که مرورگر آن را فرستاده و با نرم افزار Wireshark ضبط شده است. از روی متن به پرسش ها پاسخ دهید.

```
GET /cs453/index.html HTTP/1.1<cr><lf>Host: gai
a.cs.umass.edu<cr><lf>User-Agent: Mozilla/5.0 (
Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec
ko/20040804 Netscape/7.2 (ax) <cr><lf>Accept:ex
t/xml, application/xml, application/xhtml+xml, text
/html;q=0.9, text/plain;q=0.8,image/png,*/*;q=0.5
<cr><lf>Accept-Language: en-us,en;q=0.5<cr><lf>Accept-
Encoding: zip,deflate<cr><lf>Accept-Charset: ISO
-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr>
<lf>Connection:keep-alive<cr><lf><cr><lf>
```

الف. URL مربوط به سند در خواستی (requested document) چیست؟

<http://Gaia.cs.umass.edu/cs453/index.html>

دقت شود که آدرس URL شامل آدرس host و آدرس فایل مورد نظر بر روی سرور است. پاسخی به فرم [Gaia.cs.umass.edu/cs453/index.html](http://Gaia.cs.umass.edu/cs453/index.html) یا [Gaia.cs.umass.edu](http://Gaia.cs.umass.edu) ناقص است.

ب. چه نسخه ای از HTTP بر روی مرورگر اجرا شده است؟

HTTP/1.1

پ. مرورگر از ارتباط persistent استفاده کرده است یا non-persistent؟

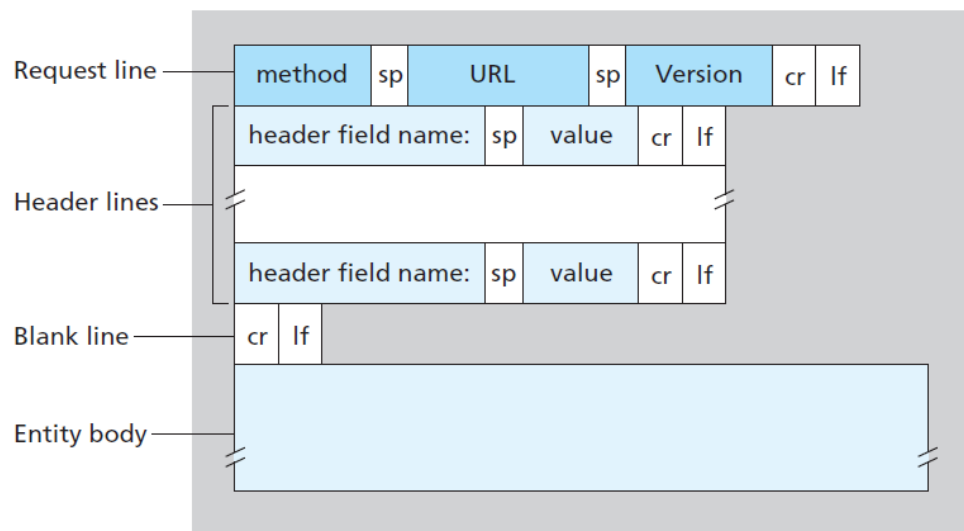
Persistent : عبارت connection: keep-alive در خط آخر این موضوع را نشان می دهد.

\* یادآوری: در ارتباط persistent تمامی object ها توسط یک ارتباط TCP از سرور به کاربر انتقال می یابد، اما در non-persistent برای هر شیء یک ارتباط مجزا ساخته می شود. این ارتباط ها بنا به تنظیمات مرورگر می تواند همزمان یا پشت سر هم باشد.

ت. آیا نشانی اینترنتی (IP Address) مربوط به کامپیوتری که مرورگر بر روی آن اجرا شده، در این متن یافت می شود؟

خیر

ث. مثال بالا را با شکل زیر که قالب کلی پیغام GET است تطبیق دهید. (تمرین اضافی، نیاز به تحویل نیست.)



۲- متن زیر پاسخی ست که سرور به پیام HTTP GET ارسال شده در پرسش قبلی فرستاده است. به پرسش ها پاسخ دهید و مشخص کنید که هر پاسخ از کدام قسمت از متن به دست آمده است.

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 07 Mar 2008
12:39:45GMT<cr><lf>Server: Apache/2.0.52 (Fedora)
<cr><lf>Last-Modified: Sat, 10 Dec2005 18:27:46
GMT<cr><lf>ETag: "526c3-f22-a88a4c80"<cr><lf>Accept-
Ranges: bytes<cr><lf>Content-Length: 3874<cr><lf>
Keep-Alive: timeout=max=100<cr><lf>Connection:
Keep-Alive<cr><lf>Content-Type: text/html; charset=
ISO-8859-1<cr><lf><cr><lf><!doctype html public "-
//w3c//dtd html 4.0 transitional//en"><lf><html><lf>
<head><lf> <meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1"><lf> <meta
name="GENERATOR" content="Mozilla/4.79 [en] (Windows NT
5.0; U) Netscape]"><lf> <title>CMPSCI 453 / 591 /
NTU-ST550A Spring 2005 homepage</title><lf></head><lf>
<much more document text following here (not shown)>
```

الف. آیا سرور سند درخواستی را پیدا کرده است؟ پاسخ در چه زمانی داده شده است؟

بله با توجه به 200 OK - ۷ مارس ۲۰۰۸ ساعت ۱۲:۳۹:۴۵ GMT

ب. آخرین باری که این سند تغییر کرده چه زمانی بوده است؟

Last-Modified: Sat, 10 Dec2005 18:27:46 GMT

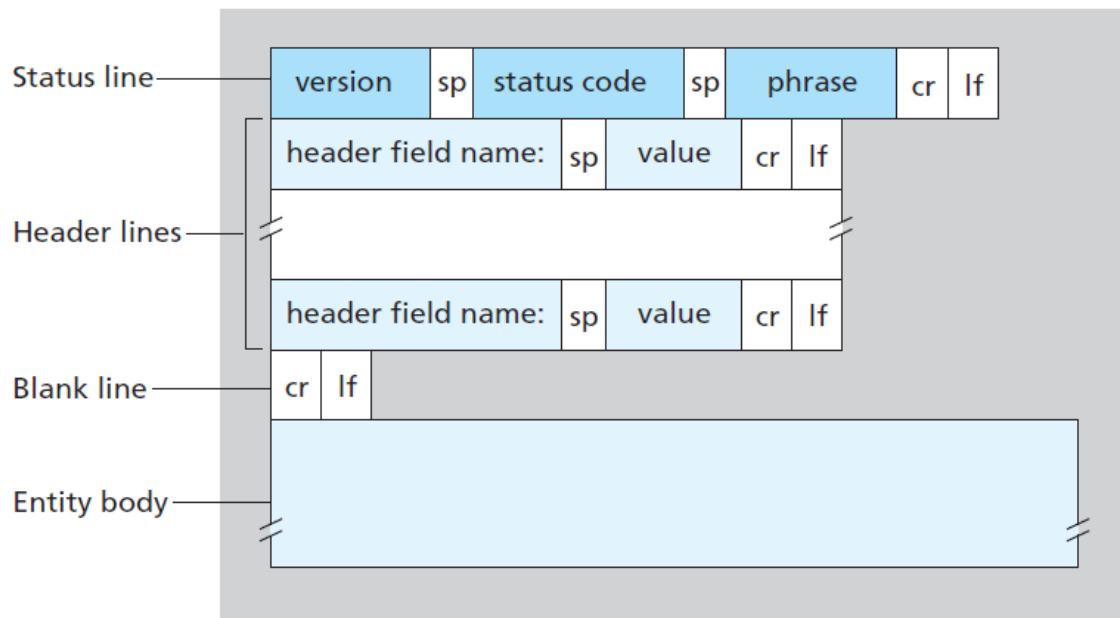
پ. این سند حاوی چند بایت بوده است و اولین پنج بایت آن چیست؟

۳۸۷۴ بیت، <!doc

ت. آیا سرور با درخواست ارتباط persistent موافقت کرده است؟

بله، چون همانند پیام قبلی نوع ارتباط در آن keep-alive ذکر شده است.

ث. مثال بالا را با شکل زیر که قالب کلی پیغام RESPONSE است تطبیق دهید. (تمرین اضافی، نیاز به تحویل نیست).



\*\*\*

۳- یک پیوند کوتاه ده متری را در نظر بگیرید که هر یک از فرستنده ها در دو طرف آن می توانند با نرخ ۱۵۰ بیت در ثانیه داده ارسال کنند. بسته های داده را هم ۱۰۰۰۰۰ بیتی و بسته های کنترلی (مثل Ack) را هم ۲۰۰ بیتی در نظر بگیرید. همچنین فرض کنید که N ارتباط موازی هر کدام  $1/N$  پهنای باند را به خود اختصاص داده اند. حال پروتکل HTTP را در نظر گرفته و فرض کنید هر شیء (object) در آن ۱۰۰ کیلوبایت است و اولین شیء آن شامل ۱۰ شیء رجوع داده شده است. (توضیح: یک شیء اصلی داریم و ۱۰ شیء که توسط آن شیء رجوع داده شده اند. پس مجموعاً ۱۱ شیء باید دریافت شود).

الف. آیا دانلود های موازی به صورت تعدادی ارتباط همزمان non-persistent HTTP توجیه پذیر است؟

ب. آیا HTTP persistent نسبت به مدل non-persistent افزایش بهره ی قابل ملاحظه ای دارد؟

در مد Non-persistent پهنای باند (= ۱۵۰ بیت در ثانیه) بین ۱۰ شیء (که برای هر کدام یک ارتباط TCP ساخته می شود)، تقسیم می شود، پس برای هر ارتباط ۱۵ bps داریم.

می دانیم برای هر ارتباط TCP سه بار تبادل داده های کنترلی لازم است. (تا عمل handshaking برای شروع ارتباط انجام شود). پس داریم:

$$T_{\text{handshaking}} = 3 * (200/150 + T_p)$$

دقت شود که در این جا هنوز در مرحله ی درخواست ارتباط با شیء اصلی (حاوی ده شیء رجوع داده شده) هستیم و لذا از کل پهنای باند استفاده می شود. همچنین می توان تاخیر انتشار ( $T_p$ ) را در مقایسه با تاخیر ارسال ناچیز در نظر گرفت. (فاصله خیلی کوتاه است، مهندسی فکر کنید!)

با در نظر گرفتن این که ارسال اولین شیء  $100000/150$  زمان می برد، زمان لازم برای دریافت این شیء برابر است با ۶۷۰,۷ ثانیه.

حال به ارتباط های شیء های رجوع داده شده می پردازیم. مراحل محاسبه مانند قبل است، منتها نرخ ارسال به ۱۵ کاهش می یابد. عدد ۶۷۰۶,۷ تاخیر ارسال این اشیاء به صورت موازی است.

در نتیجه تاخیر کل برابر ۷۳۷۷,۴ می باشد.

حال به مد persistent می پردازیم. در این جا تنها یک بار عمل handshaking انجام می شود. تاخیر مربوط به ایجاد ارتباط همانند حالت قبل ۴ ثانیه است. حال ۱۱ شیء داریم (یک شیء اصلی و ده شیء رجوع داده شده) که پشت سر هم و با کل پهنای باند انتقال می یابند. برای هر کدام نیز یک تاییدیه ارسال می شود. ( $200/150 * 11$ ) پس کل تاخیر برای دریافت کل صفحه ی درخواستی ۷۳۵۱ ثانیه است. این مقدار کم تر از یک درصد با مقدار حالت قبل تفاوت دارد و این نشان می دهد که در این مثال ارتباط persistent مزیت چندانی نسبت parallel non-persistent ندارد.

۴- تفاوت میان MAIL FROM: در پروتکل SMTP و From: در خود ایمیل را بیان کنید.

MAIL FROM: پیغامی است که از طرف SMTP Client می آید که فرستنده را مشخص می کند. اما From: بخشی از متن ایمیل است.

\*\*\*

۵- توضیح دهید که هر کدام از پروتکل های SMTP و HTTP چگونه پایان بدنه ی یک پیام را مشخص می سازند؟ آیا HTTP می تواند از همان روش SMTP استفاده نماید؟

SMTP: از یک نقطه در یک سطر مجزا استفاده می کند. (ممکن است این موضوع به ذهنتان بیاید که اگر خود متن نقطه داشت چه می شود؟ چون قبل و بعد از این نقطه ی پایانی <cr><lf> داریم، این مسئله احتمالاً برطرف می شود.)

HTTP: در سرآیند (header) بخشی وجود دارد که طول پیام را مشخص می کند.

در SMTP کاراکترها ASCII بوده اما در HTTP ساختار داده از نوع باینری است؛ در نتیجه نمی توان از روش SMTP برای HTTP استفاده کرد. (مقدار باینری هر چیزی می تواند باشد.)

نکته اضافی: منحصر بودن کاراکترهای SMTP به ASCII یک عیب به شمار می آید، چرا که در کاربردهای امروزی می خواهیم فایل های صدا، تصویر و ... را هم به متن اضافه کنیم و این ها ذاتاً به فرم باینری هستند.

حمیدرضا مازندرانی