COMP90007 Internet Technologies

Lab1

Ruixi Huo

- Email: huor@student.unimelb.edu.au
- Wechat: 76922824

- If you have any questions about lecture or workshop, pls put it on Discussion Board.

- Workshop
  - Tuesday 9:00-10:00 am Engineering B - 102
  - Thursday 11:00-12:00 am Alan Gilbert - 111

Review

A unit of information at each layer

Note that we are going to use ‘packet’ as a general term for captured entities here

Installation

Wireshark
http://www.wireshark.org/download. html.

wget
‘wget’ is a tool to fetch web resources.

A standalone Windows version can be found at
https://eternallybored.org/misc/wget/

On Linux and OS X, ‘wget’ is usually pre-installed

Lab Task
Step 1

Close all unnecessary background programs.

This includes web browsers, e-mail clients and any other programs which may generate background network traffic, which can skew the measurements you take later on.

Lab Task
Step2

Open up a **Command Prompt** and ensure that ‘wget.exe’ is in the current working directory. *An easy way to open up a Command Prompt is to Shift+Right-click in the directory you have downloaded ‘wget’ to and click Open command window here.*

In the command prompt, enter **wget www.google.com.au**

But do not press Enter yet. We need to prepare Wireshark to capture the traffic.

Do not press Enter yet

Lab Task

Step3 Open Wireshark and Prepare to start a new capture

The Wireshark Network Analyzer
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Options... Ctrl+K
Start Ctrl+E
Stop Ctrl+E
Restart Ctrl+R
Capture Filters...
Refresh Interfaces F5
Apply a display filter ...
...using this filter:
Enter a capture filter ...
Wireless Network Connection 5
Local Area Connection 5
Wireless Network Connection 6

Promiscuous mode tells Wireshark to capture all packets that travel through our network interface—even those not intended for your computer.

Name resolution tries to convert some of the numerical address values into a human readable format.

What if we don’t do filter, just start capturing packets?

179	8.830354	mygateway	Sias-MBP	DNS	229	Standard query response 0x2bf1 PTR 18
180	8.834192	mygateway	Sias-MBP	DNS	141	Standard query response 0xc4df No suc
181	8.834659	mygateway	Sias-MBP	DNS	139	Standard query response 0xb78d PTR 13
182	8.841551	mygateway	Sias-MBP	DNS	196	Standard query response 0x86d3 No suc
183	8.859358	mygateway	Sias-MBP	DNS	139	Standard query response 0xf3fb PTR 12
184	8.862312	mygateway	Sias-MBP	DNS	196	Standard query response 0xc567 No suc
185	9.481564	iPhone	224.0.0.251	MDNS	159	Standard query 0x0000 PTR _homekit._t
186	9.482177	fe80::1437:4a3b:87…	ff02::fb	MDNS	179	Standard query 0x0000 PTR _homekit._t
187	10.360997	Sias-MBP	216.5[illegible]20[illegible]	TCP	54	[illegible]1257 → 443 [ACK] Seq=1 Ack=1 Win=409
188	10.391824	216.58.220.98	[illegible]	TCP	[illegible]	[TCP ACKed unseen segment] 443 → 5125
189	10.401261	Telstra	[illegible]dc[illegible]t	[illegible]RP	[illegible]	Who has 10.0.0.102? Tell 10.0.0.103
190	10.401320	Sias-MBP	T[illegible]str	[illegible]R	[illegible]	10.0.0.102 is at ac:bc:32:c6:12:53
191	10.810477	10.0.0.103	10.0.0.255	UDP	158	1024 → 5001 Len=116
192	10.920205	Sias-MBP	101.37.225.114	TLSv1…	102	Application Data
193	10.941615	Sias-MBP	mygateway	DNS	86	Standard query 0x817e PTR 98.220.58.2
194	10.981686	mygateway	Sias-MBP	DNS	182	Standard query response 0x817e PTR 98
195	11.199879	Sias-MBP	216.58.220.132	TCP	54	51287 → 443 [ACK] Seq=1 Ack=1 Win=409
196	11.230746	216.58.220.132	Sias-MBP	TCP	66	[TCP ACKed unseen segment] 443 → 5128
197	11.257284	101.37.225.114	Sias-MBP	TLSv1…	98	Application Data

MESSY

Wireshark Filter

Wireshark · Capture Interfaces
Input
Output
Options
Traffic
Link-layer Header
Promiscuous
Snaplen (B)
Buffer (MB)
Monitor Mode
Capture Filter
less Network Connection 5
Ethernet
default
2
—
l Area Connection 5
Ethernet
default
2
—
tcp port http
less Net rk Connection 6
Ethernet
default
2
—
Enable promiscuous mode on all interfaces
Manage Interfaces...
Capture filter for selected interfaces:
tcp port http
Compile BPFs
Start
Close
Help

Lab Task
Step4

Start a new capture.

Almost immediately after, hit **Enter** in the command prompt to download the Google web page to your computer.

Once the download has finished, **stop the capture** in Wireshark.

Command Prompt

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\huor>wget www.google.com.au
--17:39:22--  http://www.google.com.au/
            => `index.html.1'
Connecting to www.google.com.au:80... connected!
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]

    0K .......... .                                          @   2.24 MB/s

17:39:23 (1.86 MB/s) - `index.html.1' saved [11719]


C:\Users\huor>
```

Web Page File Size = 11719 bytes

*Local Area Connection 5 (tcp port http)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.98.228.44	172.217.25.131	TCP	66	25771→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.013164	172.217.25.131	10.98.228.44	TCP	66	80→25771 [SYN, ACK] Seq=0 Ack=1 Win=60720 Len=0 MSS=1380 SACK_PERM=1 WS=256
3	0.013215	10.98.228.44	172.217.25.131	TCP	54	25771→80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
4	0.014056	10.98.228.44	172.217.25.131	HTTP	156	GET / HTTP/1.0
5	0.027807	172.217.25.131	10.98.228.44	TCP	60	80→25771 [ACK] Seq=1 Ack=103 Win=60928 Len=0
6	0.183710	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
7	0.183712	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
8	0.183717	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
9	0.183718	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
10	0.183776	10.98.228.44	172.217.25.131	TCP	54	25771→80 [ACK] Seq=103 Ack=5721 Win=66048 Len=0
11	0.184719	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
12	0.184722	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
13	0.184768	10.98.228.44	172.217.25.131	TCP	54	25771→80 [ACK] Seq=103 Ack=8581 Win=66048 Len=0
14	0.185387	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
15	0.185389	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
16	0.185424	10.98.228.44	172.217.25.131	TCP	54	25771→80 [ACK] Seq=103 Ack=11441 Win=66048 Len=0
17	0.185449	172.217.25.131	10.98.228.44	HTTP	1035	HTTP/1.0 200 OK (text/html)
18	0.185466	10.98.228.44	172.217.25.131	TCP	54	25771→80 [ACK] Seq=103 Ack=12423 Win=65024 Len=0
19	0.194243	10.98.228.44	172.217.25.131	TCP	54	25771→80 [FIN, ACK] Seq=103 Ack=12423 Win=65024 Len=0
20	0.206922	172.217.25.131	10.98.228.44	TCP	60	80→25771 [ACK] Seq=12423 Ack=104 Win=60928 Len=0

```
▷ Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▷ Ethernet II, Src: 18:60:24:ef:8d:9f (18:60:24:ef:8d:9f), Dst: CiscoInc_ba:09:00 (6c:9c:ed:ba:09:00)
▷ Internet Protocol Version 4, Src: 10.98.228.44, Dst: 172.217.25.131
▷ Transmission Control Protocol, Src Port: 25771, Dst Port: 80, Seq: 0, Len: 0
```

```
0000  6c 9c ed ba 09 00 18 60  24 ef 8d 9f 08 00 45 00   l......` $.....E.
0010  00 34 64 dd 40 00 80 06  e0 fb 0a 62 e4 2c ac d9   .4d.@... ...b.,..
0020  19 83 64 ab 00 50 1c a0  98 0c 00 00 00 00 80 02   ..d..P.. ........
0030  20 00 80 7d 00 00 02 04  05 b4 01 03 03 08 01 01    ..}.... ........
0040  04 02                                              ..
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.98.228.44	172.217.25.131	TCP	66	25771→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
2	0.013164	172.217.25.131	10.98.228.44	TCP	66	80→25771 [SYN, ACK] Seq=0 Ack=1 Win=60720 Len=0 MSS=1380 SACK_
3	0.013215	10.98.228.44	172.217.25.131	TCP	54	25771→80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
4	0.014056	10.98.228.44	172.217.25.131	HTTP	156	GET / HTTP/1.0
5	0.027807	172.217.25.131	10.98.228.44	TCP	60	80→25771 [ACK] Seq=1 Ack=103 Win=60928 Len=0
6	0.183710	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
7	0.183712	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
8	0.183717	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
9	0.183718	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
10	0.183776	10.98.228.44	172.217.25.131	TCP	54	25771→80 [ACK] Seq=103 Ack=5721 Win=66048 Len=0
11	0.184719	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
12	0.184722	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
13	0.184768	10.98.228.44	172.217.25.131	TCP	54	25771→80 [ACK] Seq=103 Ack=8581 Win=66048 Len=0
14	0.185387	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
15	0.185389	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
16	0.185424	10.98.228.44	172.217.25.131	TCP	54	25771→80 [ACK] Seq=103 Ack=11441 Win=66048 Len=0
17	0.185449	172.217.25.131	10.98.228.44	HTTP	1035	HTTP/1.0 200 OK (text/html)
18	0.185466	10.98.228.44	172.217.25.131	TCP	54	25771→80 [ACK] Seq=103 Ack=12423 Win=65024 Len=0
19	0.194243	10.98.228.44	172.217.25.131	TCP	54	25771→80 [FIN, ACK] Seq=103 Ack=12423 Win=65024 Len=0
20	0.206922	172.217.25.131	10.98.228.44	TCP	60	80→25771 [ACK] Seq=12423 Ack=104 Win=60928 Len=0

Lab Task
Step5

In Wireshark, under the File menu, export packet dissections as a CSV file.

Open this CSV with Excel.

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.98.228.44	172.217.25.131	TCP	66	25747→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.013024	172.217.25.131	10.98.228.44	TCP	66	80→25747 [SYN, ACK] Seq=0 Ack=1 Win=60720 Len=0 MSS=1380 SACK_PERM=1 WS=256
3	0.013075	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
4	0.013973	10.98.228.44	172.217.25.131	HTTP	156	GET / HTTP/1.0
5	0.027601	172.217.25.131	10.98.228.44	TCP	60	80→25747 [ACK] Seq=1 Ack=103 Win=60928 Len=0
6	0.198687	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
7	0.198690	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
8	0.198693	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
9	0.198694	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
10	0.198751	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=5721 Win=66048 Len=0
11	0.199480	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
12	0.199482	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
13	0.199528	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=8581 Win=66048 Len=0
14	0.200293	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
15	0.200295	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
16	0.200329	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=11441 Win=66048 Len=0
17	0.200353	172.217.25.131	10.98.228.44	HTTP	1004	HTTP/1.0 200 OK (text/html)
18	0.200371	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=12392 Win=65280 Len=0
19	0.208933	10.98.228.44	172.217.25.131	TCP	54	25747→80 [FIN, ACK] Seq=103 Ack=12392 Win=65280 Len=0
20	0.221646	172.217.25.131	10.98.228.44	TCP	60	80→25747 [ACK] Seq=12392 Ack=104 Win=60928 Len=0

```
▷ Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▷ Ethernet II, Src: 18:60:24:ef:8d:9f (18:60:24:ef:8d:9f), Dst: CiscoInc_ba:09:00 (6c:9c:ed:ba:09:00)
▷ Internet Protocol Version 4, Src: 4000p1l16850503.unimelb.edu.au (10.98.228.44), Dst: syd15s03-in-f3.1e100.net (172.217.25.131)
▷ Transmission Control Protocol, Src Port: 25747, Dst Port: 80, Seq: 0, Len: 0
```

```
0000  6c 9c ed ba 09 00 18 60  24 ef 8d 9f 08 00 45 00   l......` $.....E.
0010  00 34 61 33 40 00 80 06  e4 a5 0a 62 e4 2c ac d9   .4a3@... ...b.,..
0020  19 83 64 93 00 50 ae 7b  7e 96 00 00 00 00 80 02   ..d..P.{ ~.......
0030  20 00 08 30 00 00 02 04  05 b4 01 03 03 08 01 01    ..0.... ........
0040  04 02                                              ..
```

Windows: ipconfig
Mac: ifconfig

Command Prompt

```
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\huor>ipconfig

Windows IP Configuration


Wireless LAN adapter Wireless Network Connection 6:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wireless Network Connection 5:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Local Area Connection 5:

   Connection-specific DNS Suffix  . : unimelb.edu.au
   IPv4 Address. . . . . . . . . . . : 10.98.228.44
   Subnet Mask . . . . . . . . . . . : 255.255.252.0
   Default Gateway . . . . . . . . . : 10.98.228.1

C:\Users\huor>
```

Question 2:

How much traffic in bytes in total was received and transmitted in your request for the Google web page?

What is the percentage of traffic that originated from your computer,
and what is the percentage of traffic that was sent by the remote host?

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.98.228.44	172.217.25.131	TCP	66	25747→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.013024	172.217.25.131	10.98.228.44	TCP	66	80→25747 [SYN, ACK] Seq=0 Ack=1 Win=60720 Len=0 MSS=1380 SACK_PERM=1 WS=256
3	0.013075	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
4	0.013973	10.98.228.44	172.217.25.131	HTTP	156	GET / HTTP/1.0
5	0.027601	172.217.25.131	10.98.228.44	TCP	60	80→25747 [ACK] Seq=1 Ack=103 Win=60928 Len=0
6	0.198687	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
7	0.198690	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
8	0.198693	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
9	0.198694	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
10	0.198751	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=5721 Win=66048 Len=0
11	0.199480	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
12	0.199482	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
13	0.199528	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=8581 Win=66048 Len=0
14	0.200293	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
15	0.200295	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
16	0.200329	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=11441 Win=66048 Len=0
17	0.200353	172.217.25.131	10.98.228.44	HTTP	1004	HTTP/1.0 200 OK (text/html)
18	0.200371	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=12392 Win=65280 Len=0
19	0.208933	10.98.228.44	172.217.25.131	TCP	54	25747→80 [FIN, ACK] Seq=103 Ack=12392 Win=65280 Len=0
20	0.221646	172.217.25.131	10.98.228.44	TCP	60	80→25747 [ACK] Seq=12392 Ack=104 Win=60928 Len=0

```
▷ Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▷ Ethernet II, Src: 18:60:24:ef:8d:9f (18:60:24:ef:8d:9f), Dst: CiscoInc_ba:09:00 (6c:9c:ed:ba:09:00)
▷ Internet Protocol Version 4, Src: 4000p1l16850503.unimelb.edu.au (10.98.228.44), Dst: syd15s03-in-f3.1e100.net (172.217.25.131)
▷ Transmission Control Protocol, Src Port: 25747, Dst Port: 80, Seq: 0, Len: 0
```

```
0000  6c 9c ed ba 09 00 18 60  24 ef 8d 9f 08 00 45 00   l......` $.....E.
0010  00 34 61 33 40 00 80 06  e4 a5 0a 62 e4 2c ac d9   .4a3@... ...b.,..
0020  19 83 64 93 00 50 ae 7b  7e 96 00 00 00 00 80 02   ..d..P.{ ~.......
0030  20 00 08 30 00 00 02 04  05 b4 01 03 03 08 01 01    ..0.... ........
0040  04 02                                              ..
```

Question 3:

How do these sizes compare with the file size of the web page you downloaded?
Estimate the download protocol overhead, or percentage of the download bytes taken up by protocol overhead.

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.98.228.44	172.217.25.131	TCP	66	25747→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.013024	172.217.25.131	10.98.228.44	TCP	66	80→25747 [SYN, ACK] Seq=0 Ack=1 Win=60720 Len=0 MSS=1380 SACK_PERM=1 WS=256
3	0.013075	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
4	0.013973	10.98.228.44	172.217.25.131	HTTP	156	GET / HTTP/1.0
5	0.027601	172.217.25.131	10.98.228.44	TCP	60	80→25747 [ACK] Seq=1 Ack=103 Win=60928 Len=0
6	0.198687	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
7	0.198690	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
8	0.198693	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
9	0.198694	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
10	0.198751	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=5721 Win=66048 Len=0
11	0.199480	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
12	0.199482	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
13	0.199528	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=8581 Win=66048 Len=0
14	0.200293	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
15	0.200295	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
16	0.200329	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=11441 Win=66048 Len=0
17	0.200353	172.217.25.131	10.98.228.44	HTTP	1004	HTTP/1.0 200 OK (text/html)
18	0.200371	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=12392 Win=65280 Len=0
19	0.208933	10.98.228.44	172.217.25.131	TCP	54	25747→80 [FIN, ACK] Seq=103 Ack=12392 Win=65280 Len=0
20	0.221646	172.217.25.131	10.98.228.44	TCP	60	80→25747 [ACK] Seq=12392 Ack=104 Win=60928 Len=0

total traffic in bytes

```
▷ Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▷ Ethernet II, Src: 18:60:24:ef:8d:9f (18:60:24:ef:8d:9f), Dst: CiscoInc_ba:09:00 (6c:9c:ed:ba:09:00)
▷ Internet Protocol Version 4, Src: 4000p1l16850503.unimelb.edu.au (10.98.228.44), Dst: syd15s03-in-f3.1e100.net (172.217.25.131)
▷ Transmission Control Protocol, Src Port: 25747, Dst Port: 80, Seq: 0, Len: 0

0000  6c 9c ed ba 09 00 18 60  24 ef 8d 9f 08 00 45 00   l......` $.....E.
0010  00 34 61 33 40 00 80 06  e4 a5 0a 62 e4 2c ac d9   .4a3@... ...b.,..
0020  19 83 64 93 00 50 ae 7b  7e 96 00 00 00 00 80 02   ..d..P.{ ~.......
0030  20 00 08 30 00 00 02 04  05 b4 01 03 03 08 01 01    ..0.... ........
0040  04 02                                              ..
```

Command Prompt

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\huor>wget www.google.com.au
--17:39:22--  http://www.google.com.au/
           => `index.html.1'
Connecting to www.google.com.au:80... connected!
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]

    0K .......... .                                          @   2.24 MB/s

17:39:23 (1.86 MB/s) - `index.html.1' saved [11719]


C:\Users\huor>
```

Web Page File Size = 11719 bytes

Question 4:

What are the protocol overheads used for? Why are they useful? Aren’t they a waste of space?

Protocol Hierarchies (3)

- Example information flow supporting virtual communication in layer 5

The overheads/headers are used to allow each layer in the networking architecture perform their particular service.

These headers are part of how data is transferred between the networking layers, what we call ”encapsulation”, and keeps networking functions separate and modular between the layers.

Arguably, the size of the information contained in the headers can be reduced by having a ”monolithic” architecture to networking, where there is only one ”layer”, but then modularity and abstraction are sacrificed, which is bad from an engineering perspective.

Question 5

Are you able to identify other packets apart from the HTTP GET request and response packets in your capture? What might these packets be used for?

HTTP GET: request packets
HTTP 200 OK: response packets
SYN and SYN ACK packets: at the beginning which establishes a reliable connection between two hosts
The packets in the middle: are the data that make up the rest of the actual web page
FIN and FIN ACK packets: the packets at the end when the connection is closed.

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.98.228.44	172.217.25.131	TCP	66	25747→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.013024	172.217.25.131	10.98.228.44	TCP	66	80→25747 [SYN, ACK] Seq=0 Ack=1 Win=60720 Len=0 MSS=1380 SACK_PERM=1 WS=256
3	0.013075	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
4	0.013973	10.98.228.44	172.217.25.131	HTTP	156	GET / HTTP/1.0
5	0.027601	172.217.25.131	10.98.228.44	TCP	60	80→25747 [ACK] Seq=1 Ack=103 Win=60928 Len=0
6	0.198687	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
7	0.198690	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
8	0.198693	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
9	0.198694	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
10	0.198751	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=5721 Win=66048 Len=0
11	0.199480	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
12	0.199482	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
13	0.199528	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=8581 Win=66048 Len=0
14	0.200293	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
15	0.200295	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
16	0.200329	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=11441 Win=66048 Len=0
17	0.200353	172.217.25.131	10.98.228.44	HTTP	1004	HTTP/1.0 200 OK (text/html)
18	0.200371	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=12392 Win=65280 Len=0
19	0.208933	10.98.228.44	172.217.25.131	TCP	54	25747→80 [FIN, ACK] Seq=103 Ack=12392 Win=65280 Len=0
20	0.221646	172.217.25.131	10.98.228.44	TCP	60	80→25747 [ACK] Seq=12392 Ack=104 Win=60928 Len=0

```
▷ Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▷ Ethernet II, Src: 18:60:24:ef:8d:9f (18:60:24:ef:8d:9f), Dst: CiscoInc_ba:09:00 (6c:9c:ed:ba:09:00)
▷ Internet Protocol Version 4, Src: 4000p1l16850503.unimelb.edu.au (10.98.228.44), Dst: syd15s03-in-f3.1e100.net (172.217.25.131)
▷ Transmission Control Protocol, Src Port: 25747, Dst Port: 80, Seq: 0, Len: 0
```

```
0000  6c 9c ed ba 09 00 18 60  24 ef 8d 9f 08 00 45 00   l......` $.....E.
0010  00 34 61 33 40 00 80 06  e4 a5 0a 62 e4 2c ac d9   .4a3@... ...b.,..
0020  19 83 64 93 00 50 ae 7b  7e 96 00 00 00 00 80 02   ..d..P.{ ~.......
0030  20 00 08 30 00 00 02 04  05 b4 01 03 03 08 01 01    ..0.... ........
0040  04 02                                              ..
```

Question 6

How does do the protocol overheads relate to the networking layering architecture you have seen (or will see) in class?
Is this architecture efficient?
What are the advantages of this ‘layered-cake’ architecture?
What are the disadvantages?

A non-layered architecture may be more efficient, but at the cost of flexibility and modularity.

Disadvantage of layered-cake architecture:

1. Less efficient
2. a sizable overhead if the information to be transferred needs to be split across many packets.

Advantage of layered-cake architecture:

1. modularity and abstraction
2. information hiding
3. flexibility

Question 7

Calculate the inter-arrival times (IAT) of the packets, i.e. the time between each packet arrival.

Plot the number of packets against the IAT.

What sort of distribution do you observe? You may need to capture many more HTTP packets (this time using a web browser, for example) for a proper distribution to be seen.

1	0.000000	10.98.228.44	172.217.25.131	TCP	66	25747→80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.013024	172.217.25.131	10.98.228.44	TCP	66	80→25747 [SYN, ACK] Seq=0 Ack=1 Win=60720 Len=0 MSS=1380 SACK_PER
3	0.013075	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
4	0.013973	10.98.228.44	172.217.25.131	HTTP	156	GET / HTTP/1.0
5	0.027601	172.217.25.131	10.98.228.44	TCP	60	80→25747 [ACK] Seq=1 Ack=103 Win=60928 Len=0
6	0.198687	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
7	0.198690	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
8	0.198693	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
9	0.198694	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
10	0.198751	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=5721 Win=66048 Len=0
11	0.199480	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
12	0.199482	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
13	0.199528	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=8581 Win=66048 Len=0
14	0.200293	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
15	0.200295	172.217.25.131	10.98.228.44	TCP	1484	[TCP segment of a reassembled PDU]
16	0.200329	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=11441 Win=66048 Len=0
17	0.200353	172.217.25.131	10.98.228.44	HTTP	1004	HTTP/1.0 200 OK (text/html)
18	0.200371	10.98.228.44	172.217.25.131	TCP	54	25747→80 [ACK] Seq=103 Ack=12392 Win=65280 Len=0
19	0.208933	10.98.228.44	172.217.25.131	TCP	54	25747→80 [FIN, ACK] Seq=103 Ack=12392 Win=65280 Len=0
20	0.221646	172.217.25.131	10.98.228.44	TCP	60	80→25747 [ACK] Seq=12392 Ack=104 Win=60928 Len=0