

T2-11 Trace File Analysis - Analyzing HTTP Traffic Behavior

April 2, 2008

Tony Fortunato

Sr Network Specialist | The Technology Firm

SHARKFEST '08

Foothill College

March 31 - April 2, 2008

About your Presenter

Tony Fortunato, Sr Network Specialist, The Technology Firm

Certified Fluke Networks and Wireshark Instructor

Website: www.thetechfirm.com

A Senior Network Specialist with experience in performance testing, network design, implementation, and troubleshooting LAN/WAN/Wireless networks, desktops and servers since 1989.

Tony has taught at Colleges/Universities, Network/Interop and many onsite corporate settings to thousands of analysts.

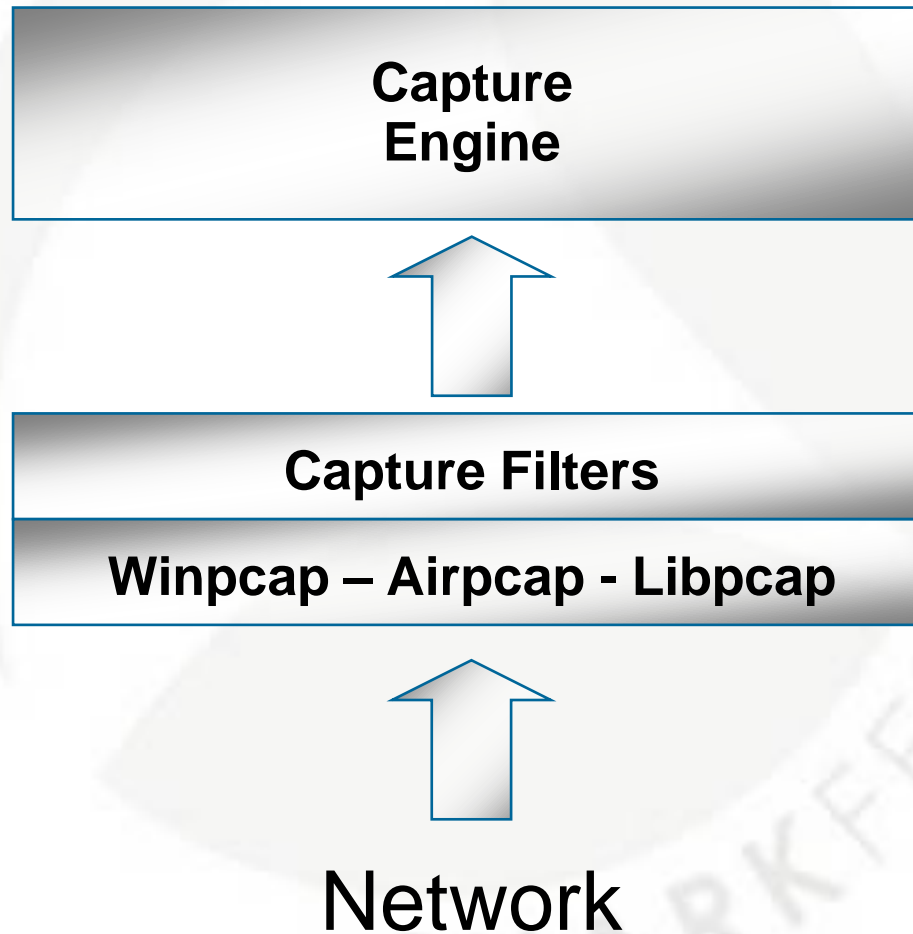
Tony is an authorized and certified Fluke Networks and Wireshark Instructor. His Pine Mountain Group CNA Level I and II certification demonstrates his vendor neutral approach to network design, support and implementations.

Tony has architected, installed and supported various types of Residential Wireless High Speed as well as hundreds of WIFI hotspots.. Tony combines custom programs, open source and commercial software to ensure a simple support infrastructure.

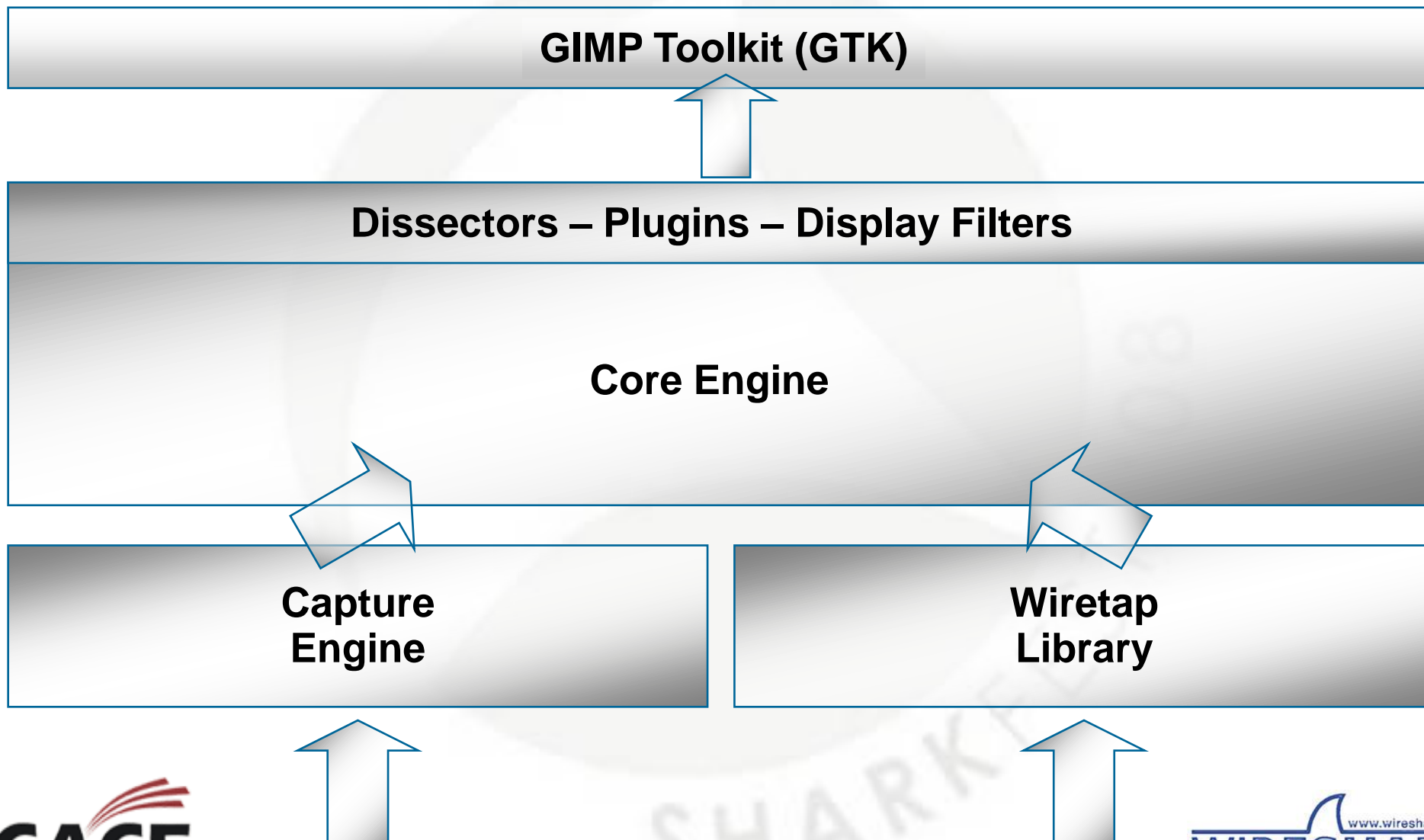
Tony works on networks from 2 to 120,000 nodes and specializes in post installation performance/design review. This process involves using various tools (Protocol analyzers, traffic generators and network management) and working on multi-vendor equipment (switches, routers, servers, etc).

Tony works at customer sites within a range of capacities from project management, network design, consulting, troubleshooting, designing customized courses and assisting with installing physical equipment.

Capturing Traffic

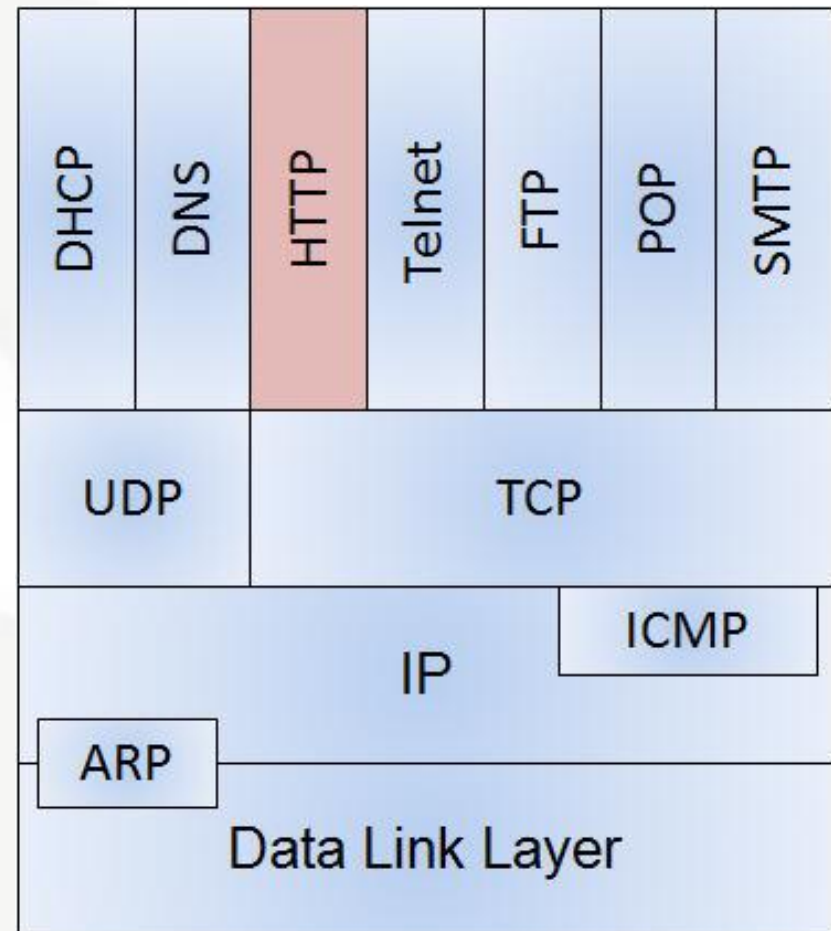


Processing Packets



Overview of HTTP

- Hypertext Transport Protocol
- RFC 2616 (HTTP v1.1)
- Distributed hypermedia information distribution application



WinInet Limits Connections Per Server

Article ID:183110 Last Review:October 26, 2007 Revision:4.2 from Microsoft support and was previously published under Q183110

WinInet is an API used for applications to use specific protocols like Gopher, FTP, and HTTP protocols to access Internet resources.

WinInet limits the number of simultaneous connections that it makes to a single HTTP server. If you exceed this limit, the requests block until one of the current connections has completed. This is by design and is in agreement with the HTTP specification and industry standards.

WinInet limits connections to a single HTTP 1.0 server to four simultaneous connections and connections to a single HTTP 1.1 server are limited to two simultaneous connections.

The HTTP 1.1 specification (RFC2616) mandates the two-connection limit. The four-connection limit for HTTP 1.0 is a self-imposed restriction that coincides with the standard that is used by a number of popular Web browsers.

You can configure WinInet to exceed this limit by creating and setting the following registry entries:

Note By changing these settings, you cause WinInet to go against the HTTP protocol specification recommendation. You should only do this if absolutely necessary and then you should avoid doing standard Web browsing while these settings are in effect: HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet Settings

MaxConnectionsPerServer REG_DWORD (Default 2)
Sets the number of simultaneous requests to a single HTTP 1.1 Server

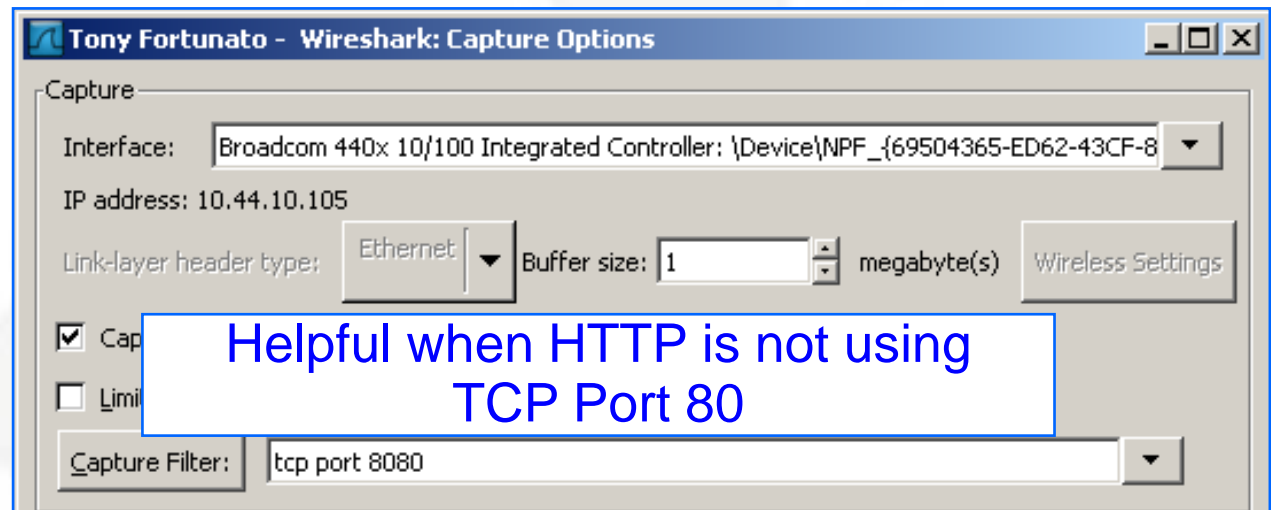
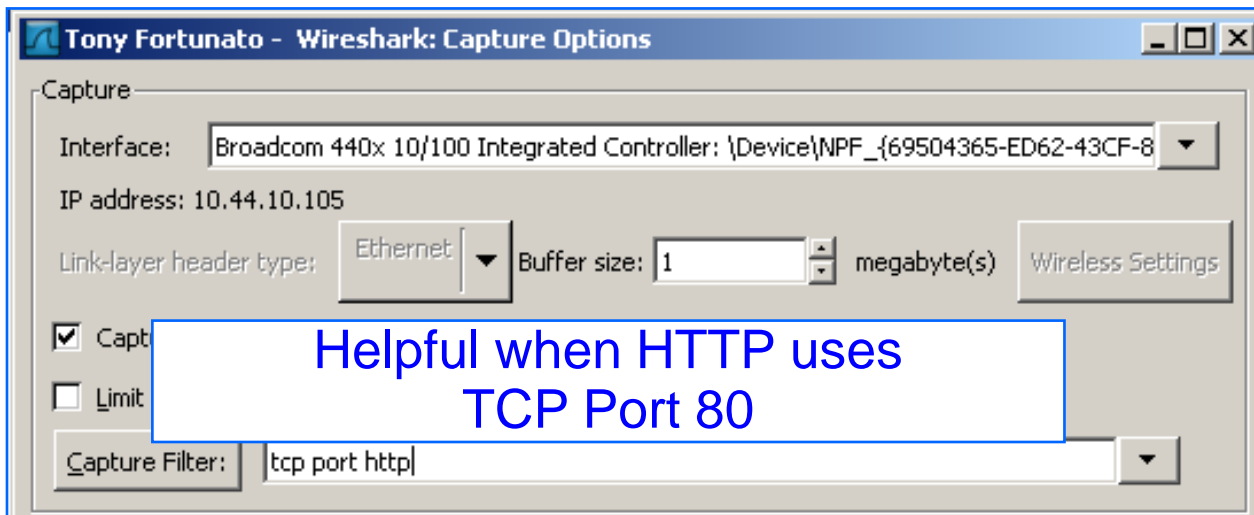
MaxConnectionsPer1_0Server REG_DWORD (Default 4)
Sets the number of simultaneous requests to a single HTTP 1.0 Server

HTTP Keep-Alives

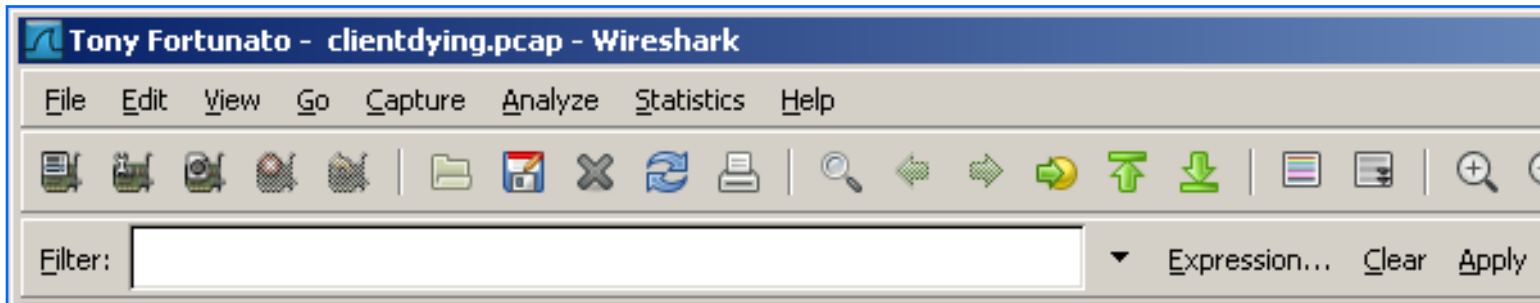
Since HTTP 1.1 uses a keep-Alive, this option is no longer required and ignored.

```
[-] Hypertext Transfer Protocol
  [-] GET / HTTP/1.1\r\n
    Host: www.thetechfirm.com\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.3) Gecko/20090519 Firefox/3.5.3\r\n
    Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/css;q=0.8,application/javascript;q=0.4,*/*;q=0.1\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Cookie: aqusr=A1044.380258BB021B29EB59\r\n
```

HTTP Capture Filter



HTTP Display Filters



You can use the following Display Filters

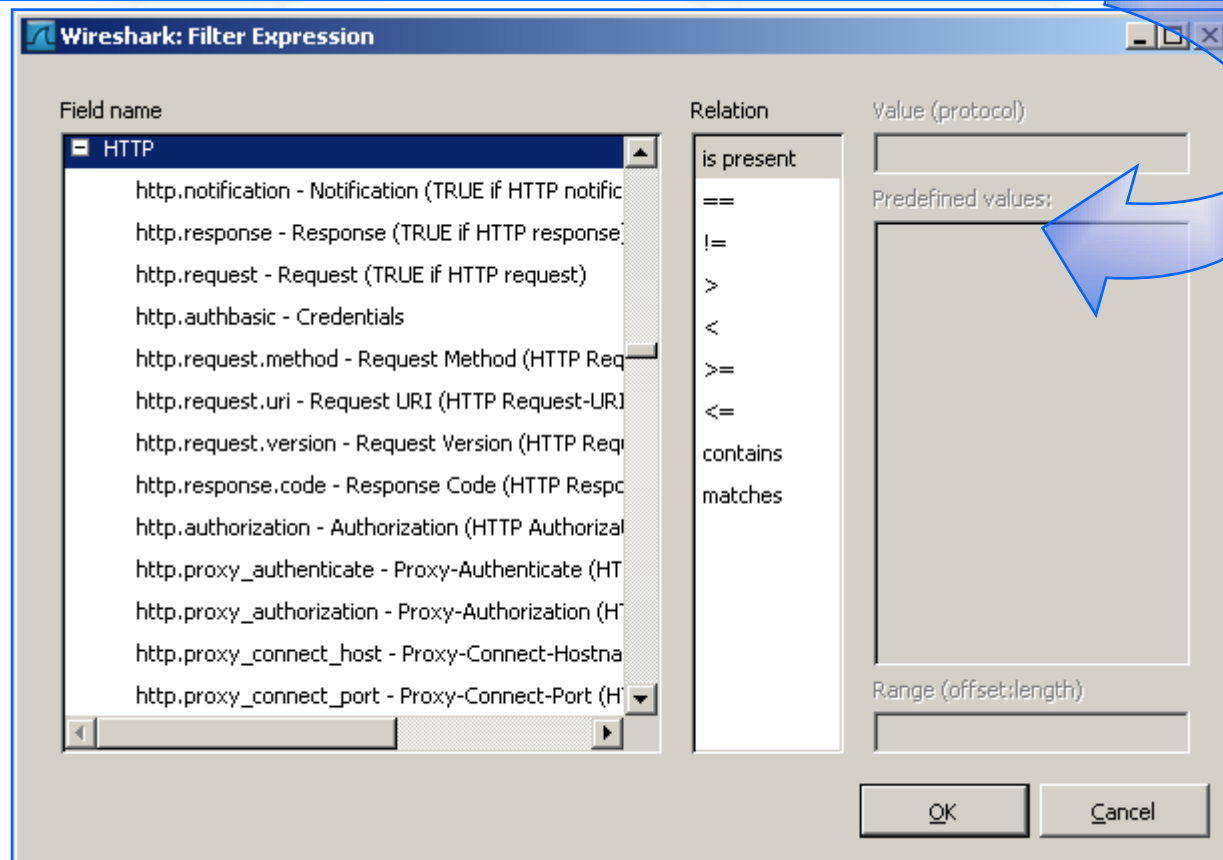
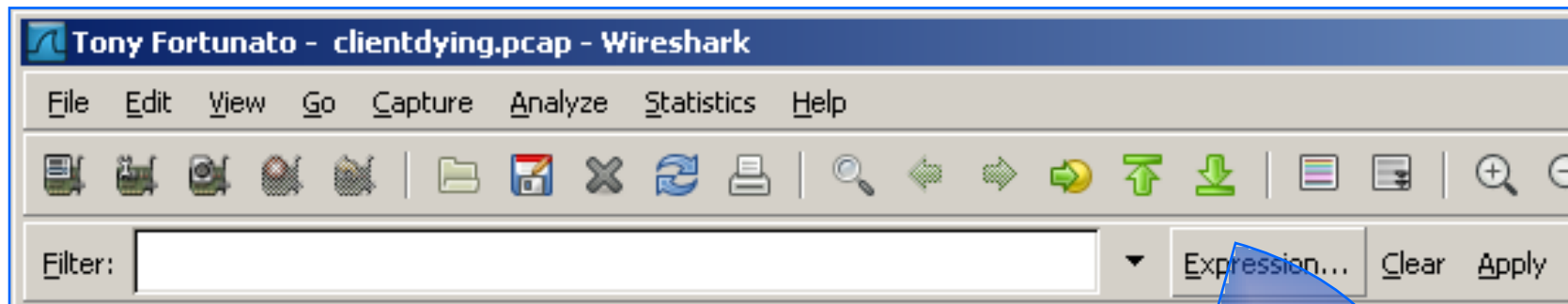
`http` (*TCP SYN, ACKs, RST or FIN packets will not be displayed*)

`tcp.port==80` (*or whatever port number you use*)

`http.request.method == "GET"`

`http.request.method == "POST"`

List of HTTP Display Fields



HTTP Packet Structure

TCP-based (Port 80)

Variable-length header

[http-post.pcap]

```
⊖ Hypertext Transfer Protocol
  ⊖ GET /apps/Agent/en-us/Agent5/chknews.asp?affid=8
    Request Method: GET
    Request URI: /apps/Agent/en-us/Agent5/chknews
    Request Version: HTTP/1.1
    Accept: */*\r\n
    User-Agent: MCUPDATE\r\n
    Host: us.mcafee.com\r\n
    \r\n
```

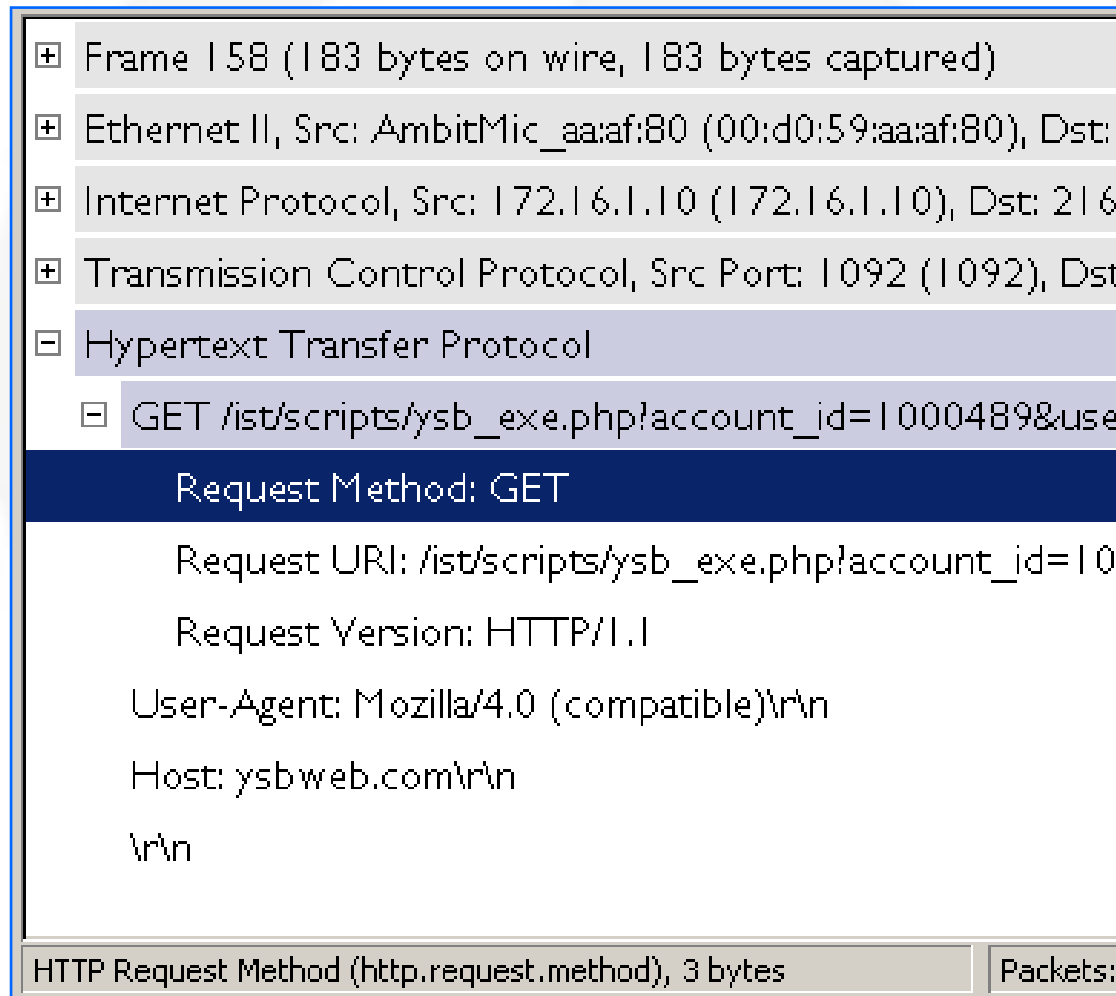
GET to read data

```
⊖ Hypertext Transfer Protocol
  ⊖ POST /apps/agent/submgr/appinstru.asp HTTP/1.1\r\n
    Request Method: POST
    Request URI: /apps/agent/submgr/appinstru.asp
    Request Version: HTTP/1.1
    Accept: */*\r\n
    ReportingSource: IDCC47C7BB09444fBA3445D72BD98CBC\r\n
    ReportingIdent: I\r\n
    Content-Type: application/binary\r\n
    User-Agent: McHttp\r\n
    Host: us.mcafee.com\r\n
    Content-Length: 526
```

POST to write data

Wireshark Field Names

As you highlight various fields in the Packet Details View, you'll see the field name displayed in the status bar.



The screenshot shows the Packet Details View in Wireshark. The tree view on the left shows the following hierarchy:

- ⊕ Frame 158 (183 bytes on wire, 183 bytes captured)
- ⊕ Ethernet II, Src: AmbitMic_aa:af:80 (00:d0:59:aa:af:80), Dst:
- ⊕ Internet Protocol, Src: 172.16.1.10 (172.16.1.10), Dst: 216
- ⊕ Transmission Control Protocol, Src Port: 1092 (1092), Dst
- ⊖ Hypertext Transfer Protocol
- ⊖ GET /ist/scripts/ysb_exe.php?account_id=1000489&use

The selected field is expanded, showing the following details:

- Request Method: GET
- Request URI: /ist/scripts/ysb_exe.php?account_id=10
- Request Version: HTTP/1.1
- User-Agent: Mozilla/4.0 (compatible)\r\n
- Host: ysbweb.com\r\n
- \r\n

The status bar at the bottom of the window displays: HTTP Request Method (http.request.method), 3 bytes | Packets:

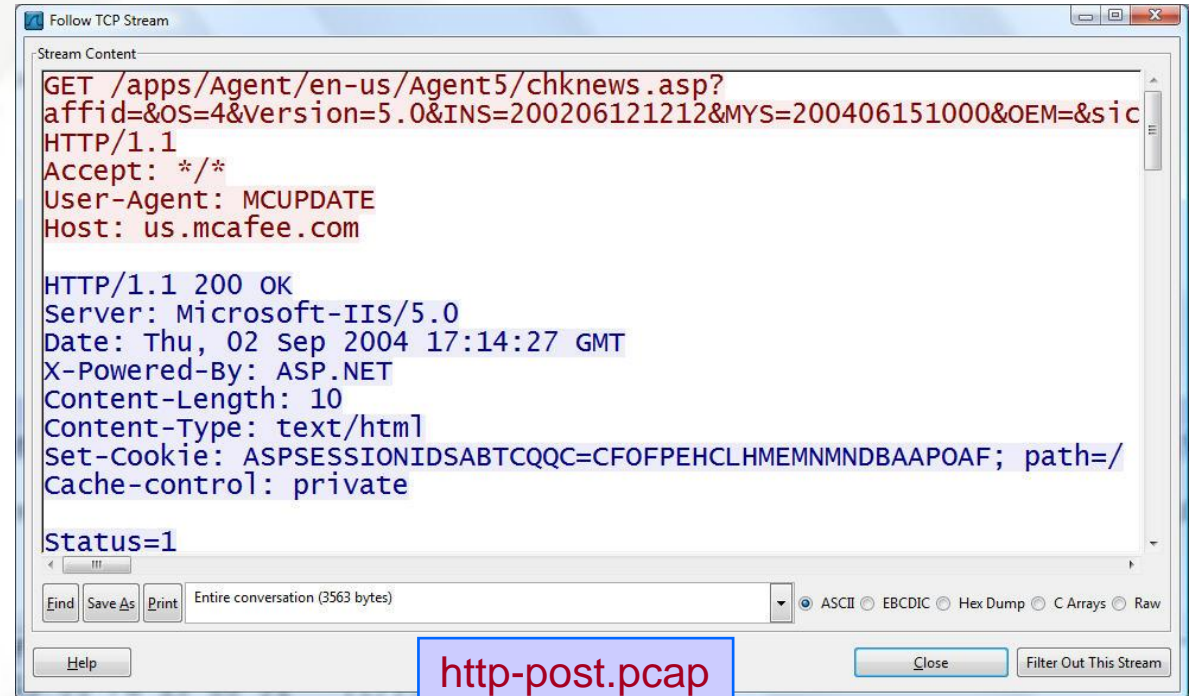
Analyze Normal HTTP Traffic

Most common commands:

- GET
- POST

Responses are numerical:

- 1xx Informational
- 2xx Successful
- 3xx Redirection
- 4xx Client error
- 5xx Server error



The screenshot shows the 'Follow TCP Stream' window in Wireshark. The 'Stream Content' pane displays the following text:

```
GET /apps/Agent/en-us/Agent5/chknews.asp?  
affid=&OS=4&version=5.0&INS=200206121212&MYS=200406151000&OEM=&sic  
HTTP/1.1  
Accept: */*  
User-Agent: MCUPDATE  
Host: us.mcafee.com  
  
HTTP/1.1 200 OK  
Server: Microsoft-IIS/5.0  
Date: Thu, 02 Sep 2004 17:14:27 GMT  
X-Powered-By: ASP.NET  
Content-Length: 10  
Content-Type: text/html  
Set-Cookie: ASPSESSIONIDSABTCQQC=CF0FPEHCLHMEMNMNDBAPOAF; path=/  
Cache-control: private  
  
Status=1
```

At the bottom of the window, there are buttons for 'Find', 'Save As', 'Print', 'Help', 'Close', and 'Filter Out This Stream'. The 'Find' button is highlighted with a blue box, and the text 'http-post.pcap' is written in red next to it.

Complete list of HTTP status Codes may be found at <http://www.iana.org/assignments/http-status-codes> or RFC2817

Dissecting HTTP GET *(http-post.pcap)*

The image shows a Wireshark packet capture snippet for an HTTP GET request. The packet is expanded to show the raw bytes and their interpretation. Blue arrows point from explanatory text on the right to specific fields in the packet structure.

```
Hypertext Transfer Protocol
  GET /apps/Agent/en-us/Agent5/
    Request Method: GET
    Request URI: /apps/Agent
    Request Version: HTTP/1.1
    Accept: */*
    User-Agent: MCUPDATE
    Host: us.mcafee.com
  
```

Command GET, POST, etc..

Uniform Resource Identifier may be classified as a locator (URL) or a name (URN) or both

What media types are Accepted

What Software made the call

HTTP Connection

The ‘*Connection;*’ default is typically *keep-alive*, but a server may request the client to close the connection with a ‘*close*’ response.

```
Hypertext Transfer Protocol
GET /coop/images/google_custom_search_smnar.gif HTTP/1.1\r\n
Host: www.google.com\r\n
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.1) Gecko/20080701 Firefox/3.0\r\n
Accept: image/png,*/*;q=0.5\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
Keep-Alive: 300\r\n
Connection: keep-alive\r\n
Referer: http://www.thetechfirm.com/\r\n
Cookie: __utma=173272373.2018293701.1186276722.11879197
```

```
Hypertext Transfer Protocol
HTTP/1.1 301 Moved Permanently\r\n
Date: Sun, 07 Jan 2007 08:23:36 GMT\r\n
Server: Apache\r\n
Location: http://espn.go.com/\r\n
Content-Length: 227
Connection: close\r\n
Content-Type: text/html; charset=iso-8859-1\r\n
\r\n
```

Dissecting HTTP Reponse *(http-post.pcap)*

```
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
    Request Version: HTTP/1.1
    Response Code: 200
    Server: Microsoft-IIS/5.0\r\n
    Date: Thu, 02 Sep 2004 17:14:27 GMT\r\n
    X-Powered-By: ASP.NET\r\n
    Content-Length: 10
    Content-Type: text/html\r\n
    Set-Cookie: ASPSESSIONID$ABTCQQC=
    Cache-control: private\r\n
    \r\n
  Line-based text data: text/html
    Status=1\r\n
    \r\n
```

Request Version: HTTP/1.1 ← HTTP Version .9, 1.0, 1,1

Content-Length: 10 ← 10 Bytes of Data

Cache-control: private ← Only receiving device can cache this data

Status=1

HTTP Commands, RFC's and Status Codes Reference

Commands:

Method	References
DELETE	RFC 1945
GET	RFC 1945
HEAD	RFC 1945
LINK	RFC 1945
OPTIONS	RFC 2068
PATCH	RFC 2068
POST	RFC 1945
PUT	RFC 1945
TRACE	RFC 2068
UNLINK	RFC 1945

Status code categories:

Category	Description
1yz	Informational.
2yz	Success.
3yz	Redirection.
4yz	Client error.
5yz	Server error.

HTTP Filter Reference

- Capture Filter
tcp port 80
- Display Filter
Status line
Expressions
Documentation

Display Filter Reference: Hypertext Transfer Protocol

Protocol field name: http

Versions: 0.10.0 to 0.99.6

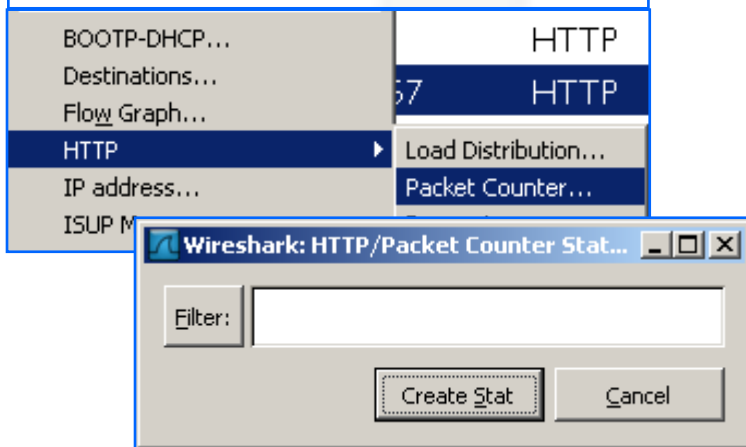
[Back to Display Filter Reference](#)

Field name	Type	Description	Versions
http.accept	String	Accept	0.10.8 to 0.99.6
http.accept_encoding	String	Accept Encoding	0.10.8 to 0.99.6
http.accept_language	String	Accept-Language	0.10.8 to 0.99.6
http.authbasic	String	Credentials	0.10.0 to 0.99.6
http.authorization	String	Authorization	0.10.0 to 0.99.6
http.cache_control	String	Cache-Control	0.10.8 to 0.99.6
http.connection	String	Connection	0.10.8 to 0.99.6
http.content_encoding	String	Content-Encoding	0.10.2 to 0.99.6
http.content_length	Unsigned 32-bit integer	Content-Length	0.10.1 to 0.99.6
http.content_type	String	Content-Type	0.10.0 to 0.99.6
http.cookie	String	Cookie	0.10.8 to 0.99.6

HTTP Packet Counter Information

A good start to analyzing HTTP is to document the number of Get commands a webpage produces

From the Statistics HTTP menu choose Packet Counter

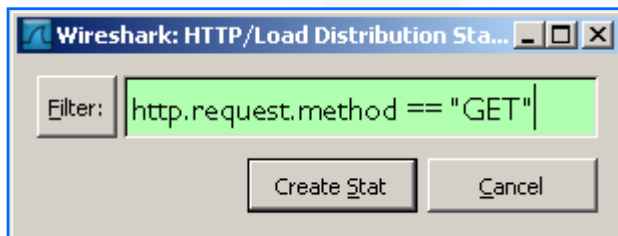
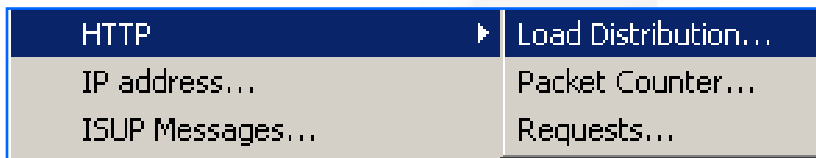


The screenshot shows the 'HTTP/Packet Counter' dialog box with a table of statistics. The table has columns for Topic / Item, Count, Rate, and Percent. The data is as follows:

Topic / Item	Count	Rate	Percent
Total HTTP Packets	244	0.006069	
HTTP Request Packets	125	0.003109	51.23%
GET	125	0.003109	100.00%
HTTP Response Packets	118	0.002935	48.36%
???; broken	0	0.000000	0.00%
1xx: Informational	0	0.000000	0.00%
2xx: Success	115	0.002861	97.46%
200 OK	115	0.002861	100.00%
3xx: Redirection	3	0.000075	2.54%
301 Moved Permanently	1	0.000025	33.33%
302 Found	2	0.000050	66.67%
4xx: Client Error	0	0.000000	0.00%
5xx: Server Error	0	0.000000	0.00%
Other HTTP Packets	1	0.000025	0.41%

Print screen is the only way to capture this information

HTTP/Load Distribution



Print screen is the only way to capture this information

A screenshot of the 'HTTP/Load Distribution' window in Wireshark. It displays a table with columns: Topic / Item, Count, Rate, and Percent. The data is as follows:

Topic / Item	Count	Rate	Percent
[-] HTTP Requests by Server	125	0.003109	
[+] HTTP Requests by Server Address	125	0.003109	100.00%
[-] HTTP Requests by HTTP Host	125	0.003109	100.00%
[+] www.espn.com	1	0.000025	0.80%
[+] espn.go.com	2	0.000050	1.60%
[+] assets.espn.go.com	91	0.002264	72.80%
[+] sports.espn.go.com	9	0.000224	7.20%
[+] scores.espn.go.com	2	0.000050	1.60%
[+] log.go.com	4	0.000099	3.20%
[+] espn-ak.starwave.com	7	0.000174	5.60%
[+] ehg-dig.hitbox.com	3	0.000075	2.40%
[+] w88.go.com	2	0.000050	1.60%
[+] static.espn.go.com	3	0.000075	2.40%
[+] seavideo-ak.espn.go.com	1	0.000025	0.80%
[+] HTTP Responses by Server Address	118	0.002935	

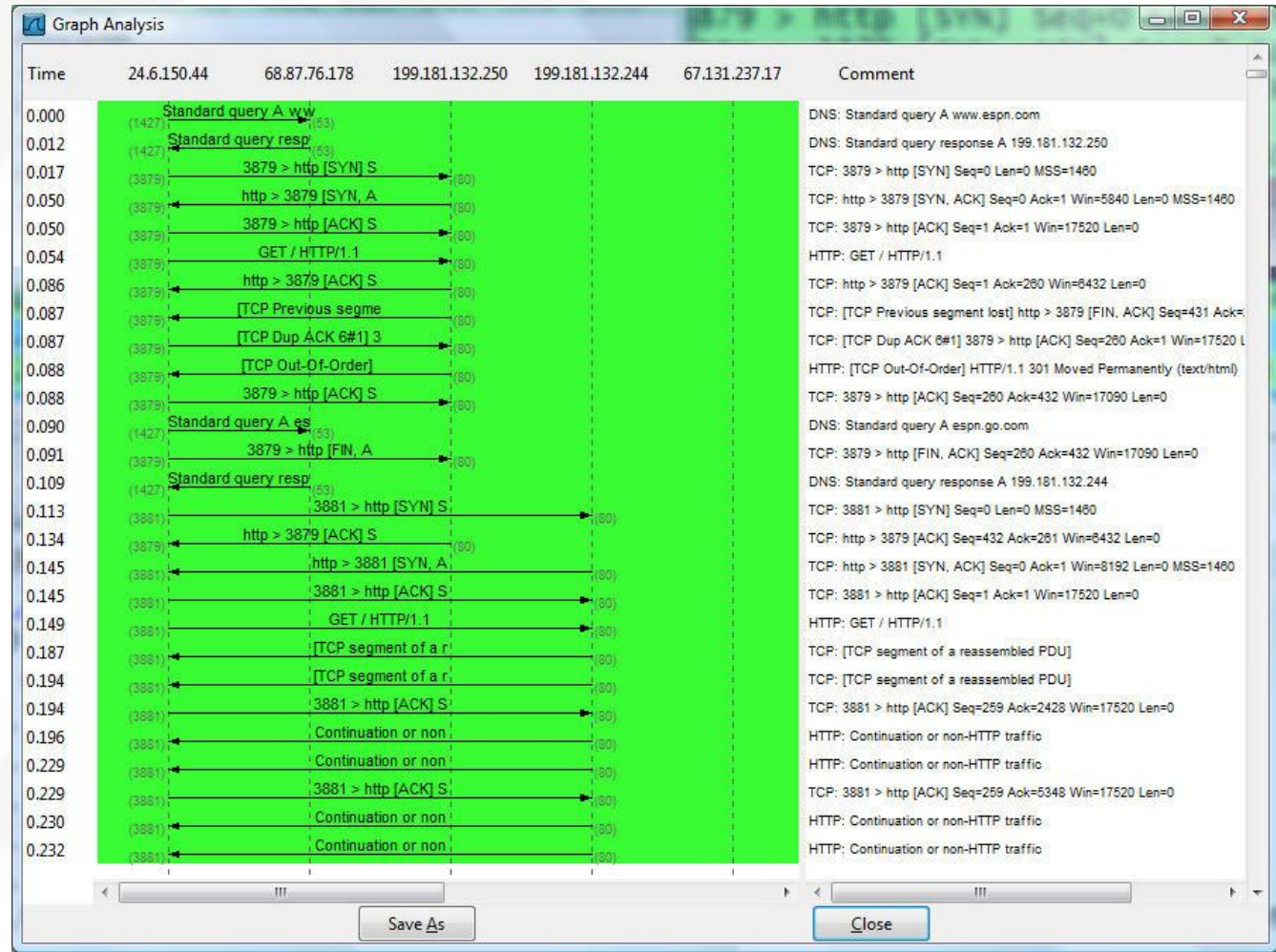
Analyze Unusual HTTP Traffic

http-espn.pcap

Notes:

Dependencies on other web sites.

Not all HTTP requests are successful.



HTTP Statistics

Load Distribution

Topic / Item	Count	Rate	Percent
[-] HTTP Requests by Server	290	0.008152	
[+] HTTP Requests by Server Address	290	0.008152	100.00%
[+] HTTP Requests by HTTP Host	290	0.008152	100.00%
HTTP Responses by Server Address	0	0.000000	

Close

Requests

Topic / Item	Count	Rate	Percent
[-] HTTP Requests by HTTP Host	290	0.008152	
[+] www.espn.com	2	0.000056	0.69%
[+] espn.go.com	60	0.001687	20.69%
[+] espn-ak.starwave.com	128	0.003598	44.14%
[+] adsatt.espn.go.com	14	0.000394	4.83%
[+] espn.starwave.com	4	0.000112	1.38%
[+] sports.espn.go.com	18	0.000506	6.21%
[+] espn-att.starwave.com	8	0.000225	2.76%
[+] ad.doubleclick.net	2	0.000056	0.69%
[+] m1.2mdn.net	2	0.000056	0.69%
[+] js.adsonar.com	1	0.000028	0.34%
[+] log.go.com	8	0.000225	2.76%
[+] static.espn.go.com	20	0.000562	6.90%
[+] adsatt.espn.starwave.com	4	0.000112	1.38%
[+] ads.espn.adsonar.com	8	0.000225	2.76%
[+] rsi.espn.go.com	4	0.000112	1.38%
[+] 3ps.go.com	2	0.000056	0.69%
[+] simg.zedo.com	1	0.000028	0.34%

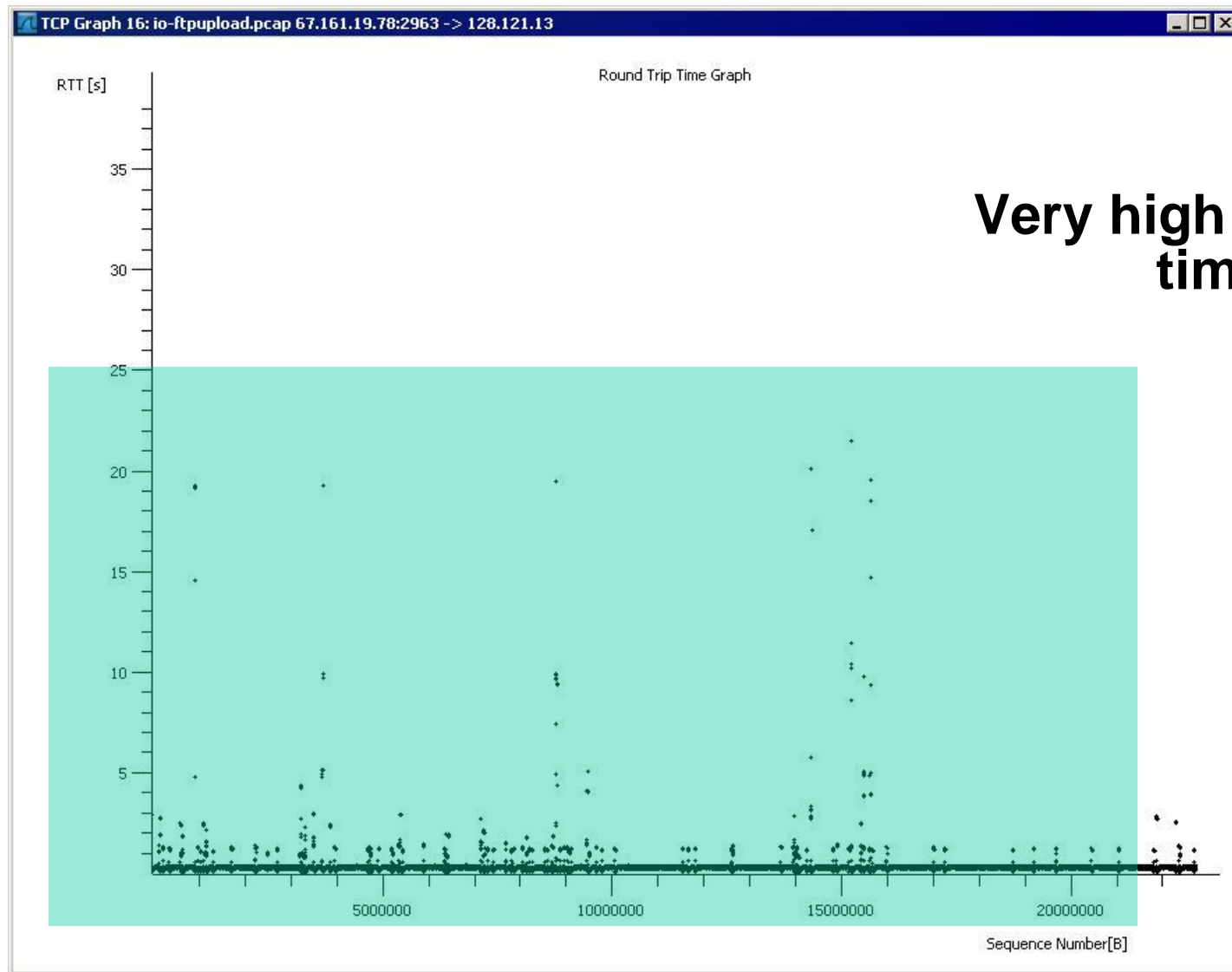
Close

Packet Counter

Topic / Item	Count	Rate	Percent
[-] Total HTTP Packets	290	0.008152	
[+] HTTP Request Packets	12	0.000337	4.14%
[+] HTTP Response Packets	278	0.007815	95.86%
Other HTTP Packets	0	0.000000	0.00%

Close

Round Trip Time Graphs



Very high round trip times!

TCP Stream Graphs

Stream Content

```
GET / HTTP/1.1
Accept: /*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR
1.0.3705; .NET CLR 1.1.4322; Media Center PC 4.0)
Cookie: userid=4592139c-00042-01ba1-6289bcc
RSP_DAEMON=4dd70aa9c3c3e0eb6b6df82a86f8def9
Connection: Keep-Alive
Host: hp-laptop.aol.com

HTTP/1.0 200 OK
X-RSP: 1
Pragma: no-cache
Cache-Control: no-cache, no-store, private, max-age=0
Expires: 0
Set-Cookie: cobr=hp-laptop.aol.com;DOMAIN=.aol.com;PATH=/
Set-Cookie: POP_COOKIE=name=a2RkMDExMTY3NTk2MTQzeA%3d%
3d;Path=/;Domain=eatps.web.aol.com;Expires=Tue, 30 Dec 2008 20:15:43 GMT
MIME-Version: 1.0
Date: Sun, 31 Dec 2006 20:15:43 GMT
Server: AOLserver/4.0.9b
Content-Type: text/html; charset=utf-8
Connection: keep-alive
Content-Encoding: gzip
Content-Length: 12133
```

client communications in red
(by default)

server communications in blue
(by default)

Find Save As Print Entire conversation (26170 bytes) ASCII EBCDIC Hex Dump C Arrays Raw

Help Close Filter Out This Stream

Decode As

If your HTTP traffic uses a different port than TCP 80, use the Analyze-> Decode As feature.

5 00 ..y..p..7@`..E.
d2@.....
0 18 jl.....7;.[P.
e ..}].GET /tony.
a 48 html HTTP/1.1..H
0 36 ost: 206.210.106

Do not decode

TCP destination (8123) port(s) as

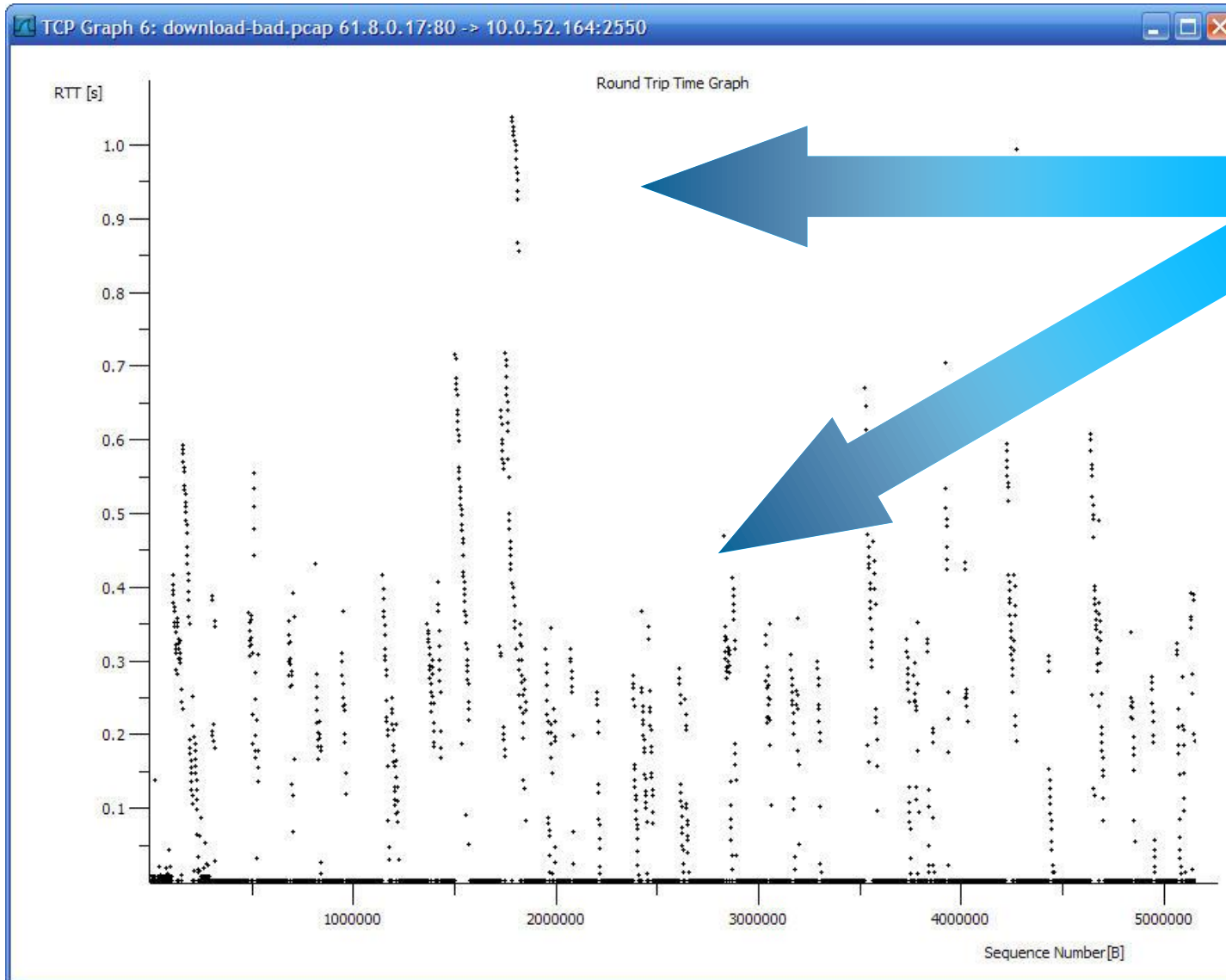
Show Current
Clear

Help OK

H.501
H248
HTTP
ICAP
iFCP
IMAP
IPDC
IRC
ISAKMP
iSCSI

TCP 2775 > 8123 [ACK] Seq=1 Ack=1
HTTP GET /tony.html HTTP/1.1

Using the Round Trip Time Graph



TCP Receiver Congestion

Wireshark: window-frozen.pcap - Wireshark

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
25	0.209709	61.8.0.17	10.0.52.164	HTTP	Continuation or non-HTTP traffic
26	0.216050	61.8.0.17	10.0.52.164	HTTP	Continuation or non-HTTP traffic
27	0.216069	10.0.52.164	61.8.0.17	TCP	2550 > http [ACK] seq=0 Ack=26280 win=730
28	0.241568	61.8.0.17	10.0.52.164	HTTP	Continuation or non-HTTP traffic
29	0.247779	61.8.0.17	10.0.52.164	HTTP	Continuation or non-HTTP traffic

Wireshark: 15 Expert Infos

Errors: 0 Warnings: 0 Notes: 3 Chats: 0 Details

Group	Protocol	Summary	Count
Sequence	TCP	Zero window	7
Packet:	30		
Packet:	32		
Packet:	34		
Packet:	36		
Packet:	38		
Packet:	40		
Packet:	42		
Sequence	TCP	Keep-Alive	6
Sequence	TCP	Window update	2

Close

Packet 36 details: TCP [TCP Zerowindow] 2550 > http [ACK] Seq=0

Packet bytes pane: ...c Port: 2550 (2550), Dst Port: http (80), Seq: 0, Ack: 29200, L

File->Export->Objects->HTTP

This feature allows you to review all the files retrieved as well as rebuilding those files

The screenshot shows the Wireshark interface with the File menu open, highlighting the Export option. The HTTP object list window is visible, showing a table of objects. The 'banner.png' object is selected, and its preview is shown in the banner.png - Windows Picture and Fax Viewer window. The 'Save All' button in the HTTP object list window is circled in blue.

Packet num	Hostname	Content Type	Offset	Filename
133	ca.my.yahoo.com	text/html		
205	us.lrd.yahoo.com	text/html		
243	www.wireshark.org	text/html		
261	www.wireshark.org	text/css		
273	www.wireshark.org	text/css		
289	www.wireshark.org	application/x-javascript	2244	common.js
312	www.wireshark.org	image/png	137	clear.png
321	www.wireshark.org	application/x-javascript	6242	menu.js
418	www.google-analytics.com	image/gif	35	&utmacc=__utma%&utmcc=__utma%&utmcc=__utma%
429	www.wireshark.org	image/png	46317	banner.png
434	www.google-analytics.com	image/gif	35	&utmacc=__utma%&utmcc=__utma%&utmcc=__utma%
437	www.google-analytics.com	image/gif	35	&utmacc=__utma%&utmcc=__utma%&utmcc=__utma%
444	www.google-analytics.com	image/gif	35	&utmacc=__utma%&utmcc=__utma%&utmcc=__utma%
447	www.wireshark.org	image/png	798	feed16.png
456	www.wireshark.org	image/png	156	nav.bg.png

Other things to consider ..

When troubleshooting, analyzing or baselining HTTP, you should monitor the following additional protocols.

- DNS
- WINS
- LDAP
- Proxy server communication

- Noting if data submitted (POSTED) to server is in clear-text or not, is very helpful.