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, Networld/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.

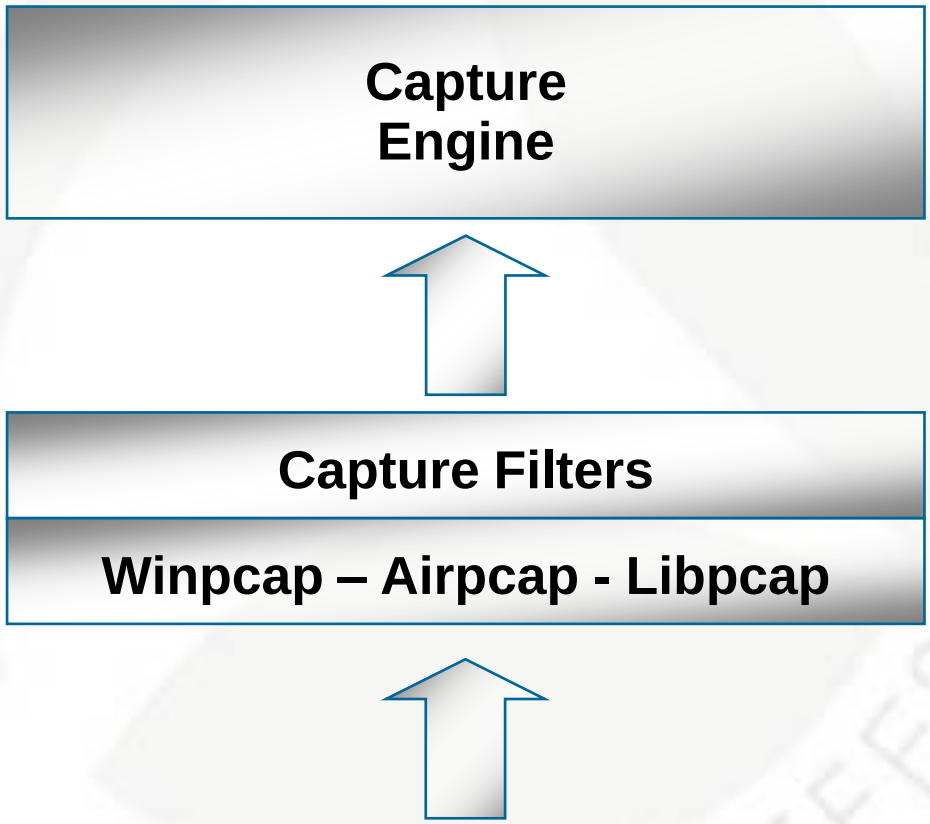
Tony works on networks from 2 to 120,000 nodes and specializes in post installation performance/design review. This process involves using various tools (Protocol analyzers, traffic generators and network management) and working on multi-vendor equipment (switches, routers, servers, etc).

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.

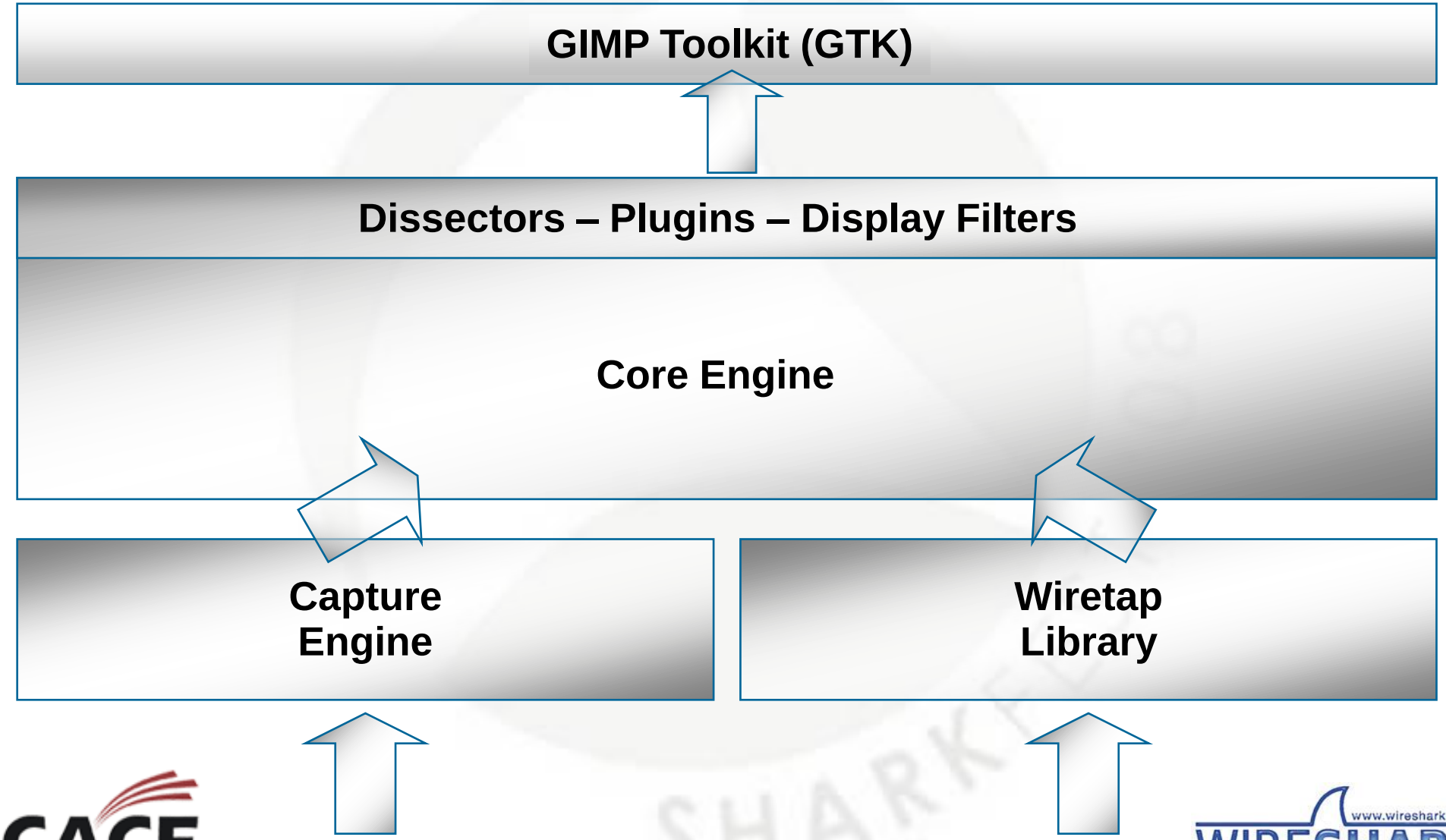
CACE TECHNOLOGIES

www.wiresharkU.com WIRESHARK UNIVERSITY

Capturing Traffic

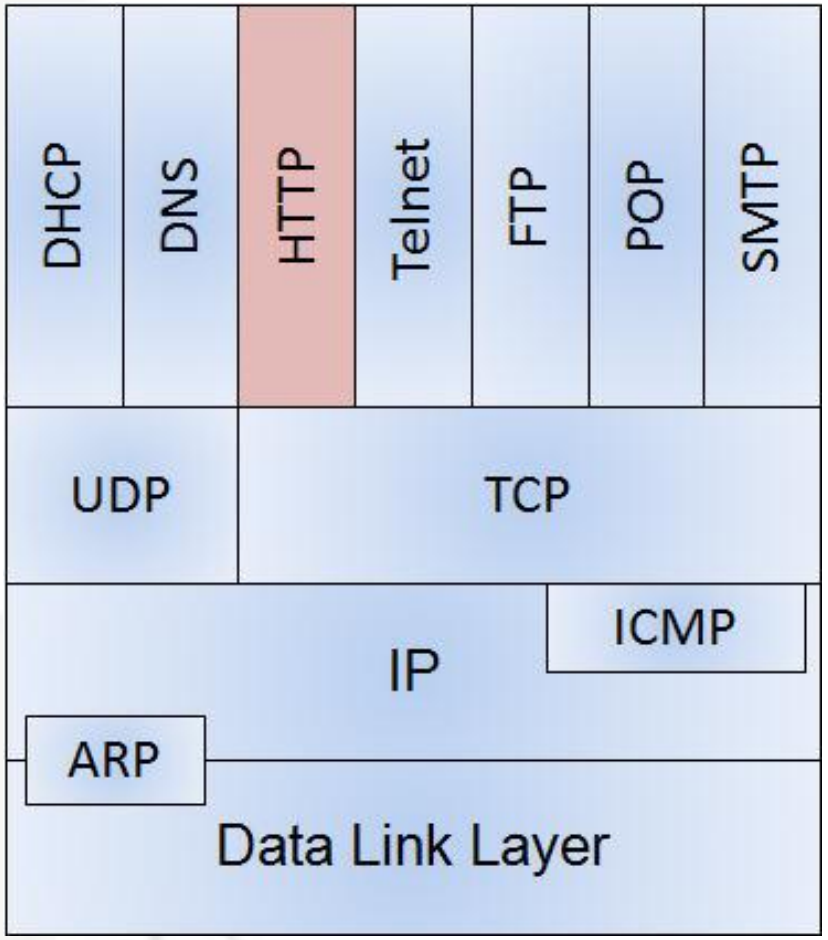


Processing Packets



Overview of HTTP

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



WinInet Limits Connections Per Server

Article ID:183110Last Review:October 26, 2007Revision: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:

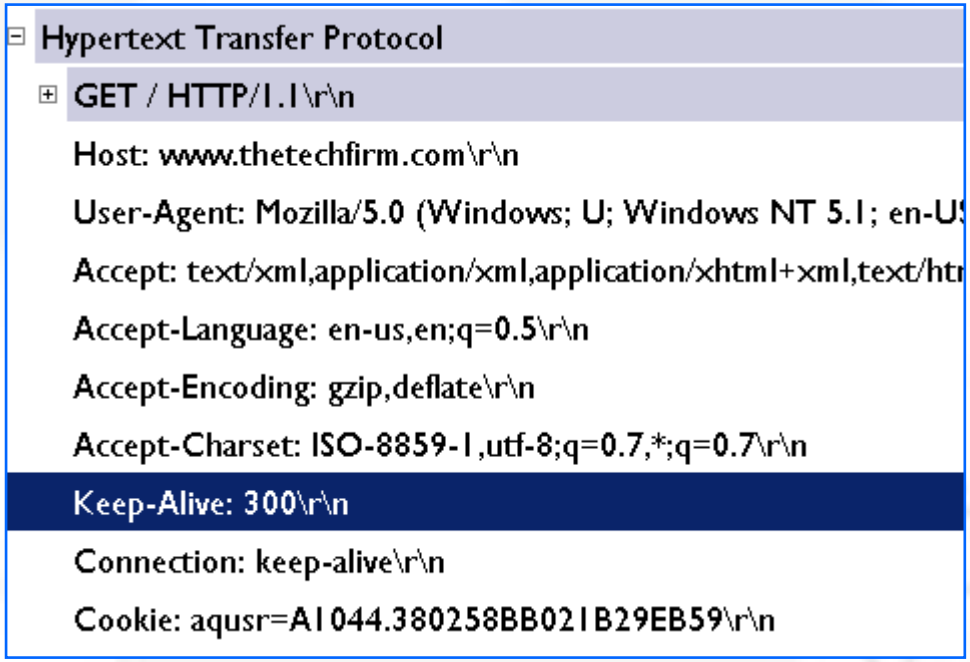
Note By changing these settings, you cause WinInet to go against the HTTP protocol specification recommendation. You should only do this if absolutely necessary and then you should avoid doing standard Web browsing while these settings are in effect:HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Internet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

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

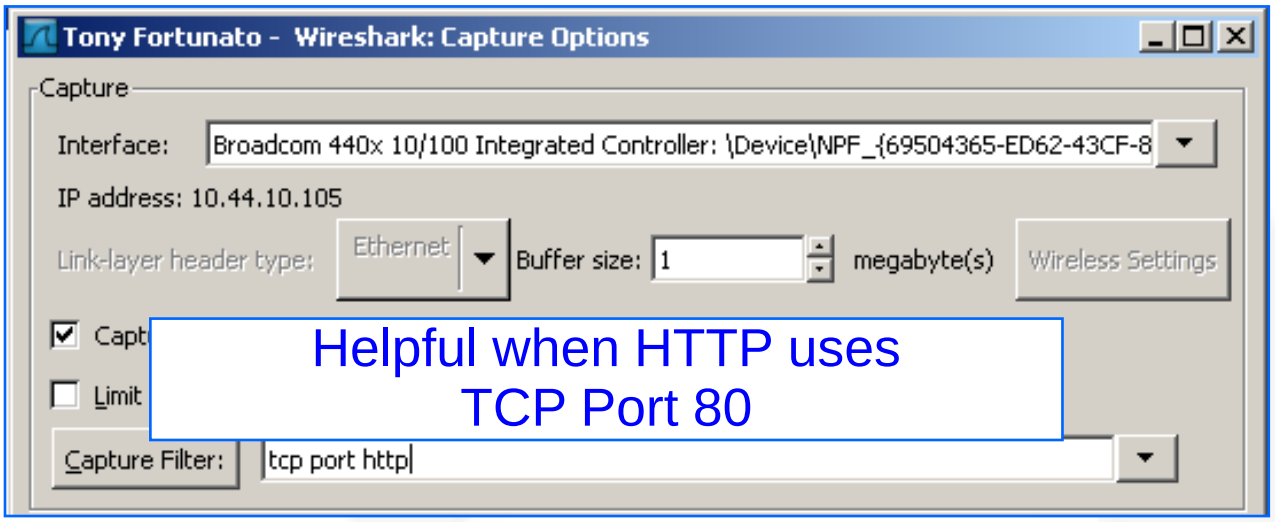
MaxConnectionsPer1_0Server REG_DWORD (Default 4)
Sets the number of simultaneous requests to a single HTTP 1.0 Server

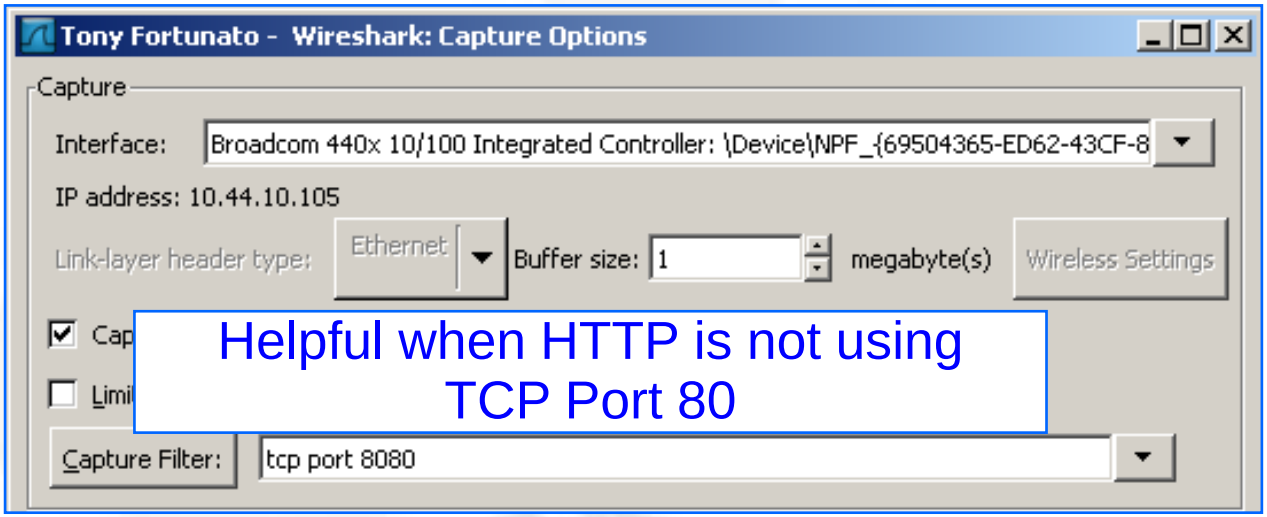
HTTP Keep-Alives

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

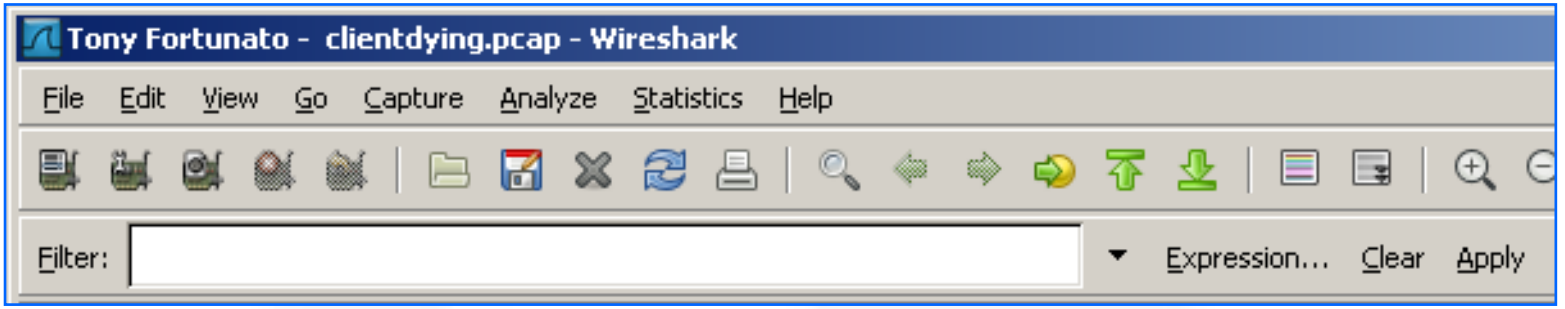


HTTP Capture Filter





HTTP Display Filters



You can use the following Display Filters

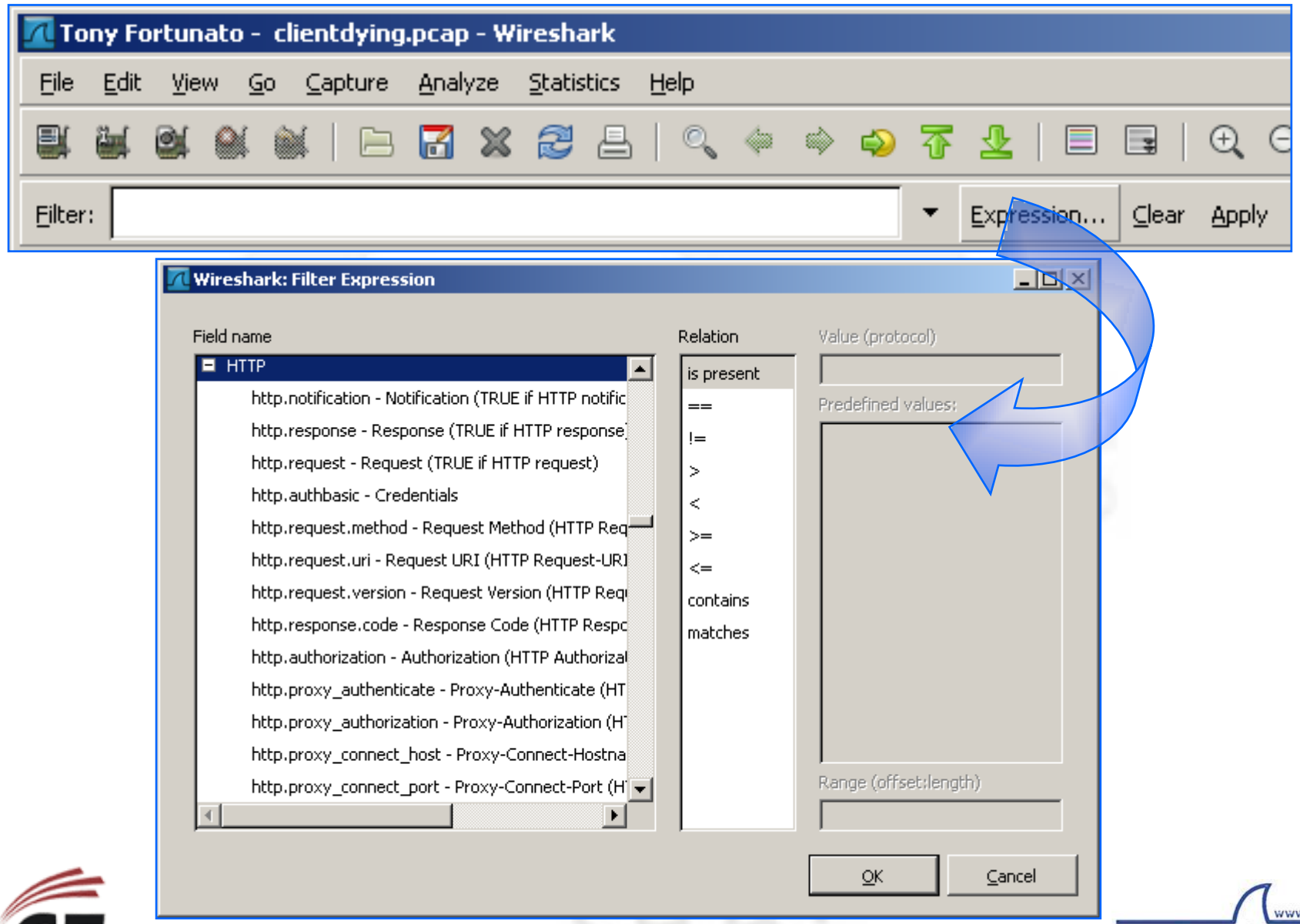
http *(TCP SYN, ACKs, RST or FIN packets will not be displayed)*

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

http.request.method == "GET“

http.request.method == “POST"

List of HTTP Display Fields



CACE TECHNOLOGIES

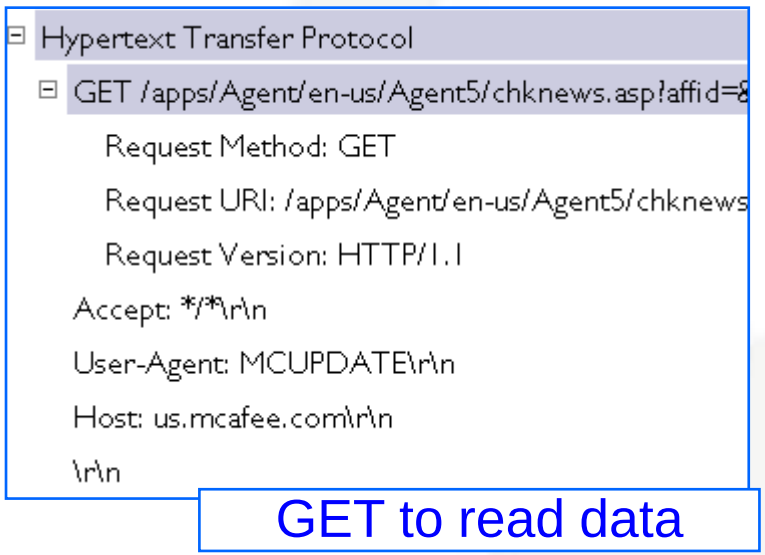
www.wiresharkU.com WIRESHARK UNIVERSITY

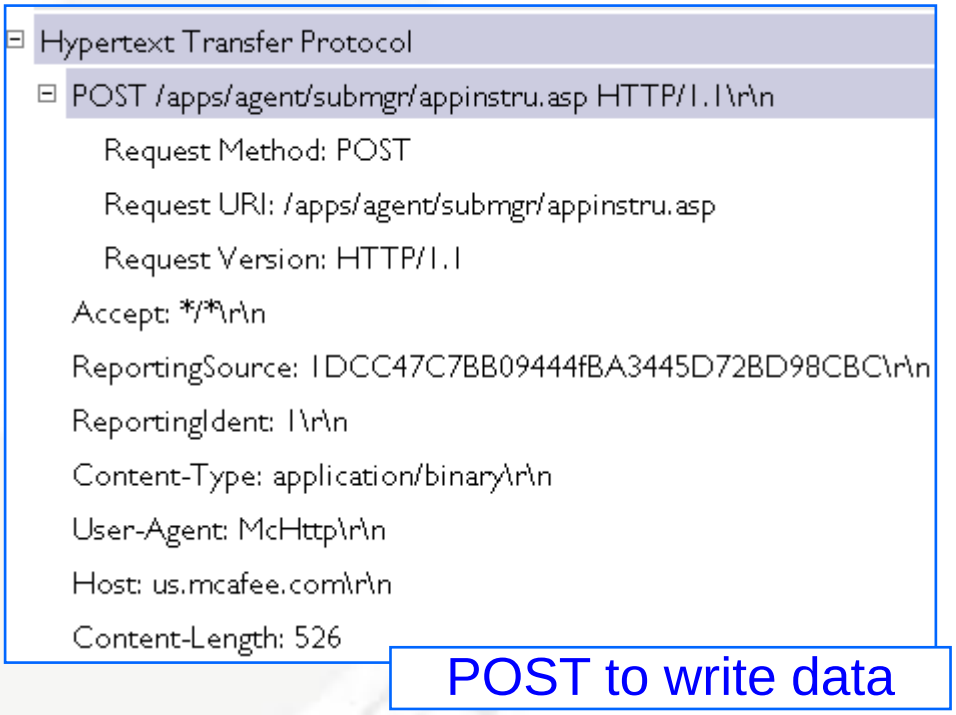
HTTP Packet Structure

TCP-based (Port 80)

Variable-length header

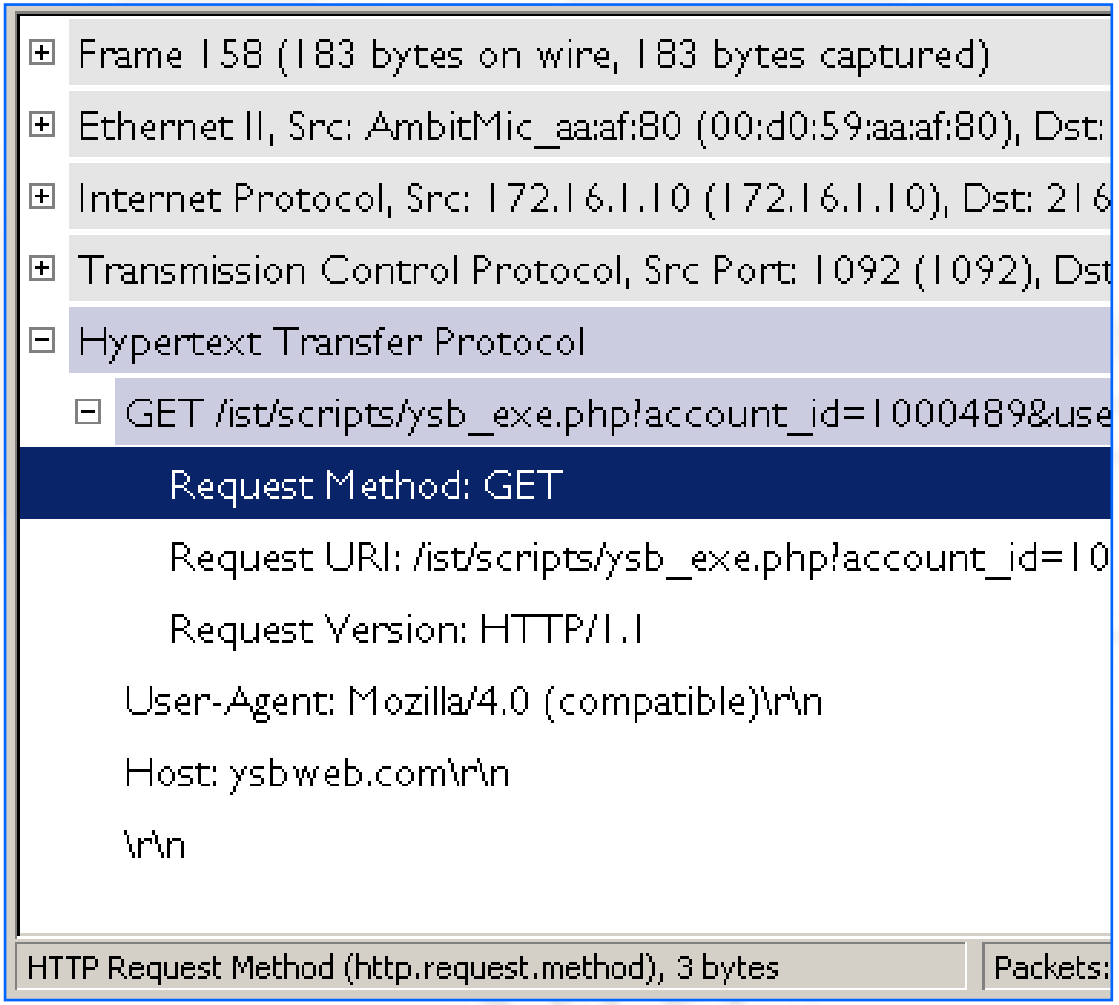
[http-post.pcap]





Wireshark Field Names

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



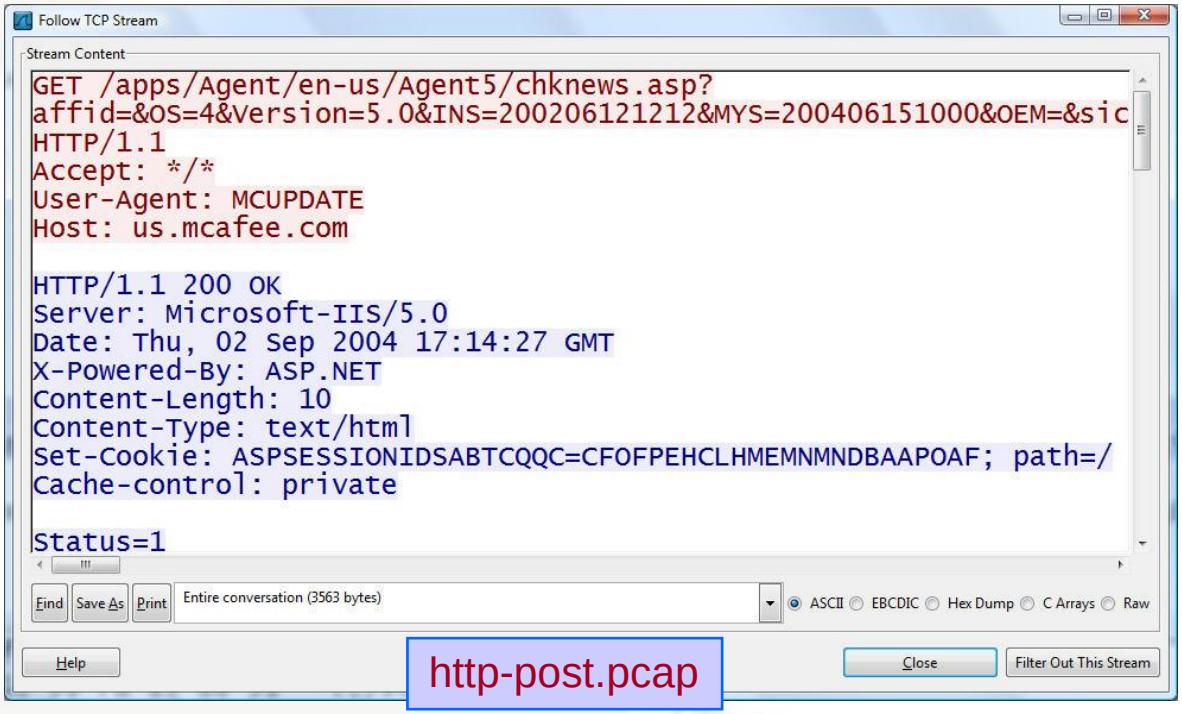
Analyze Normal HTTP Traffic

Most common commands:

- GET
- POST

Responses are numerical:

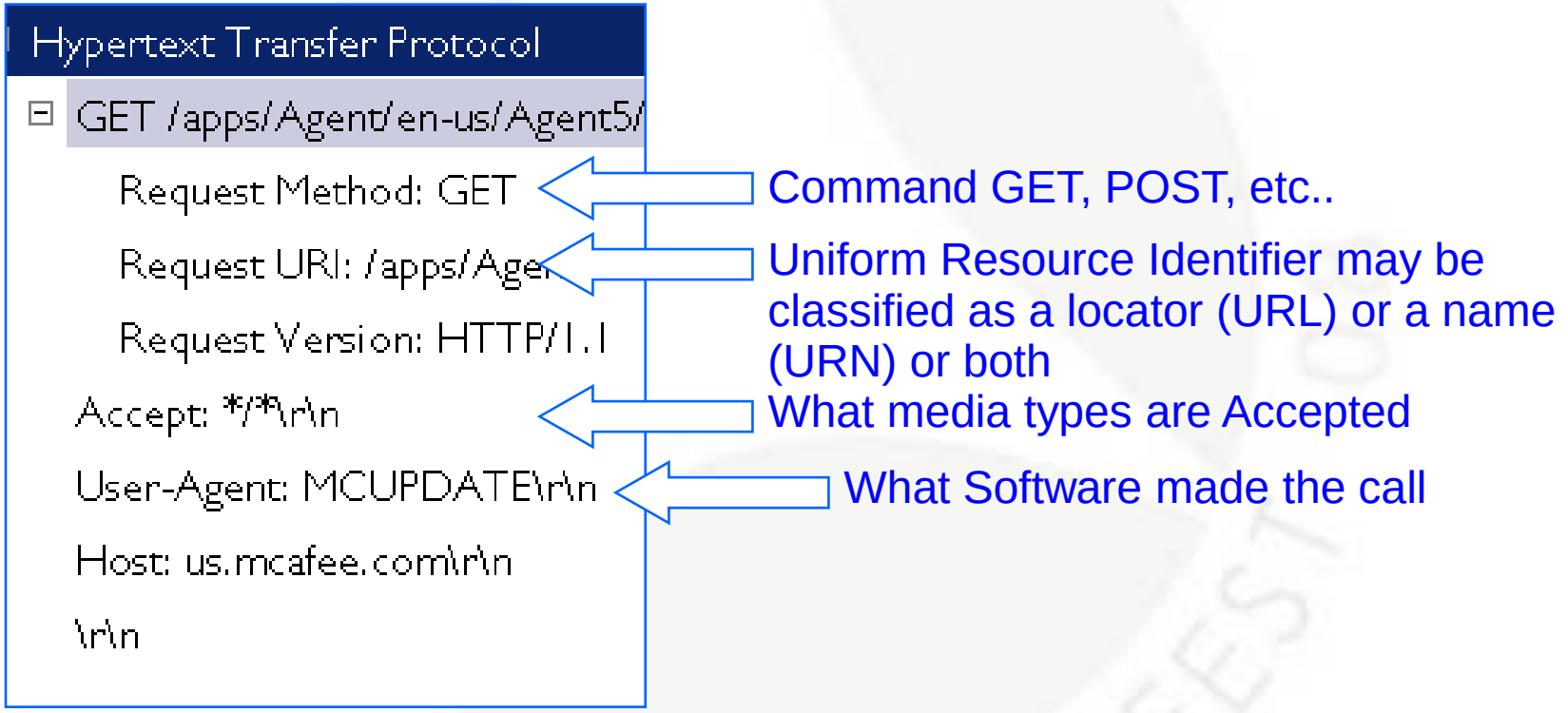
- 1xx Informational
- 2xx Successful
- 3xx Redirection
- 4xx Client error
- 5xx Server error



http-post.pcap

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)*

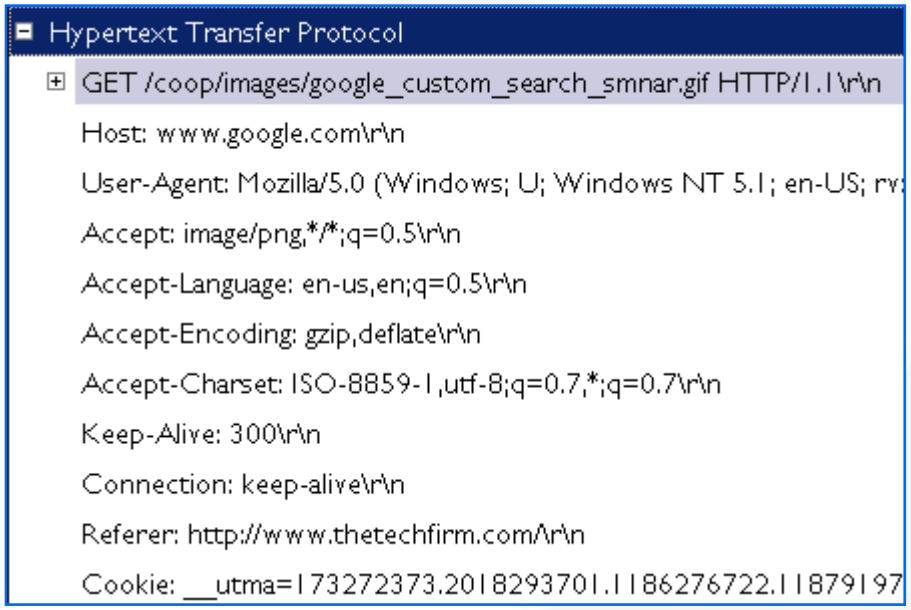


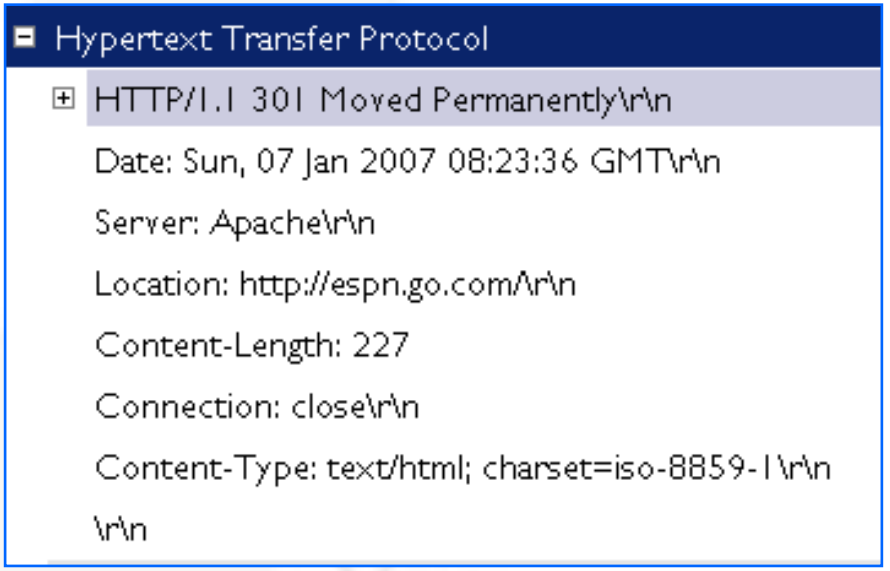
CACE TECHNOLOGIES

www.wiresharkU.com WIRESHARK UNIVERSITY

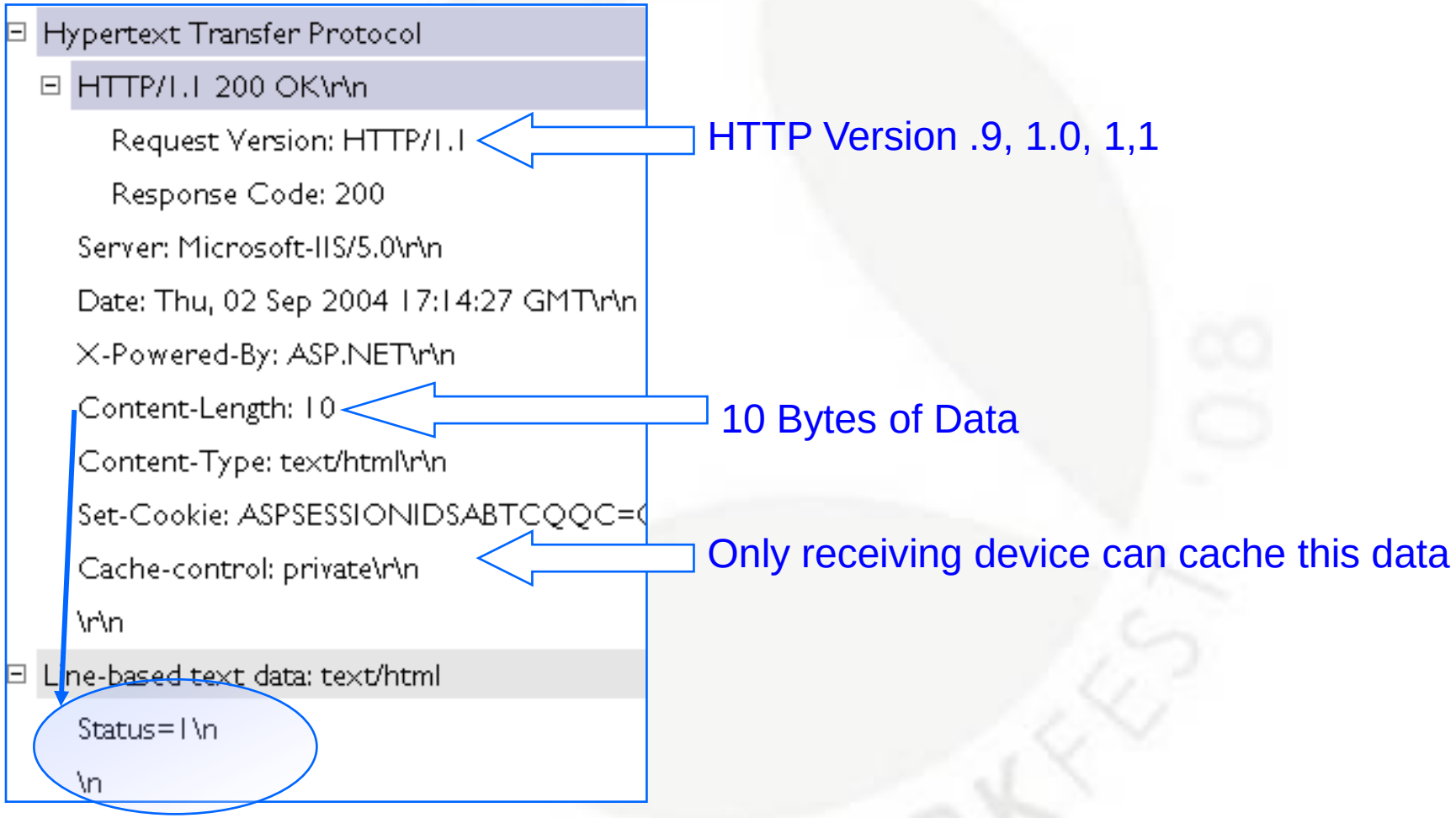
HTTP Connection

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





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



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

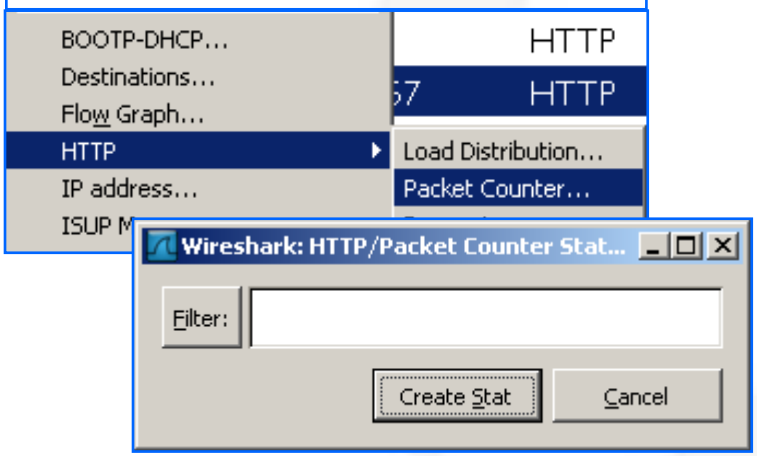
CACE TECHNOLOGIES

www.wiresharkU.com WIRESHARK UNIVERSITY

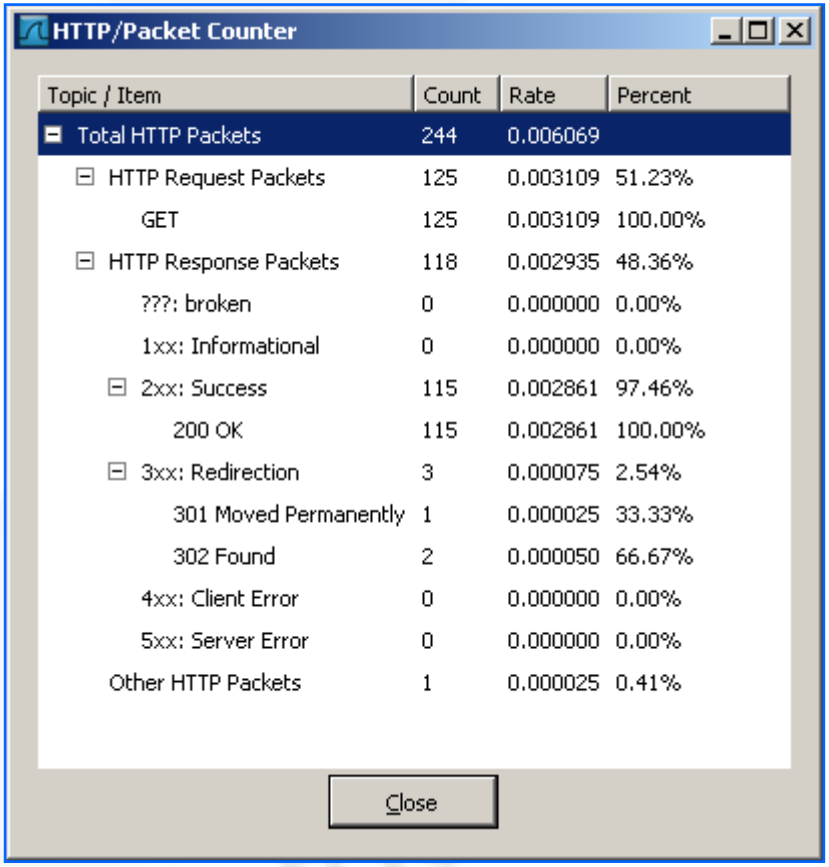
HTTP Packet Counter Information

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



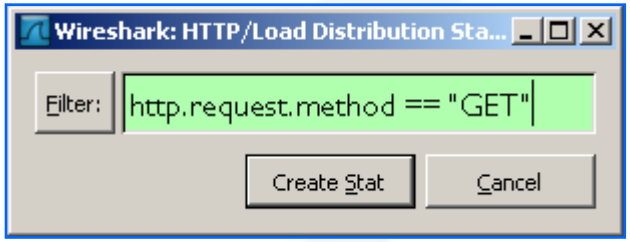
Print screen is the only way to capture this information



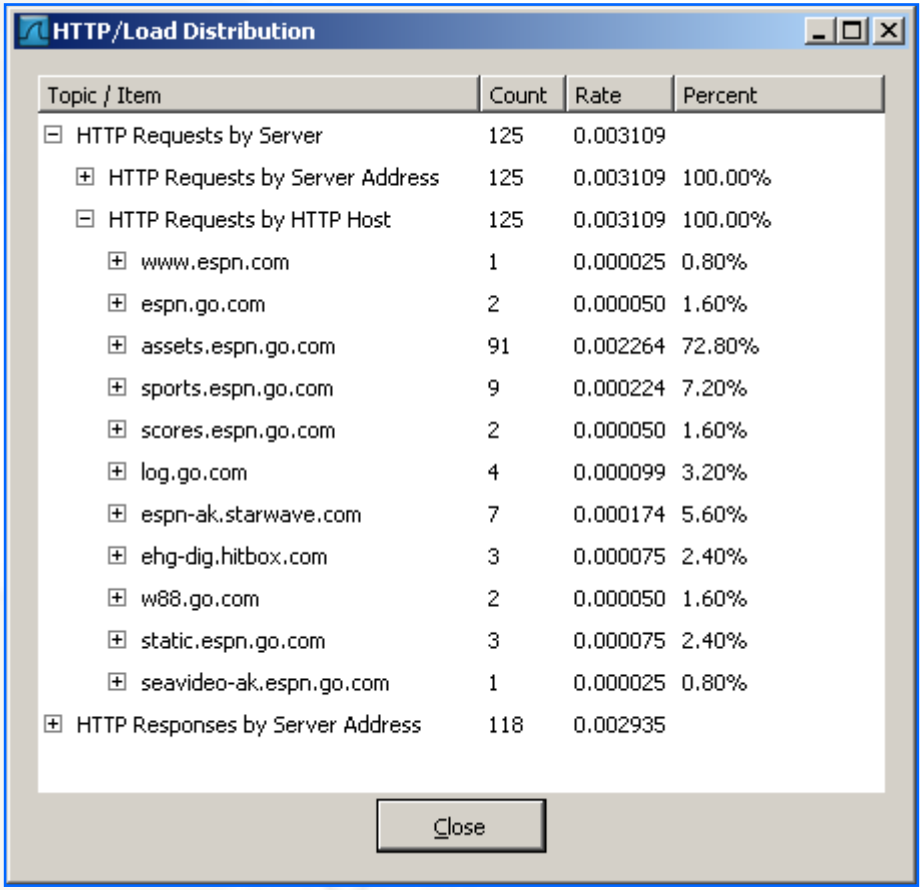
CACE TECHNOLOGIES

www.wiresharkU.com WIRESHARK UNIVERSITY

HTTP/Load Distribution



Print screen is the only way to capture this information



CACE TECHNOLOGIES

www.wiresharkU.com
WIRESHARK UNIVERSITY

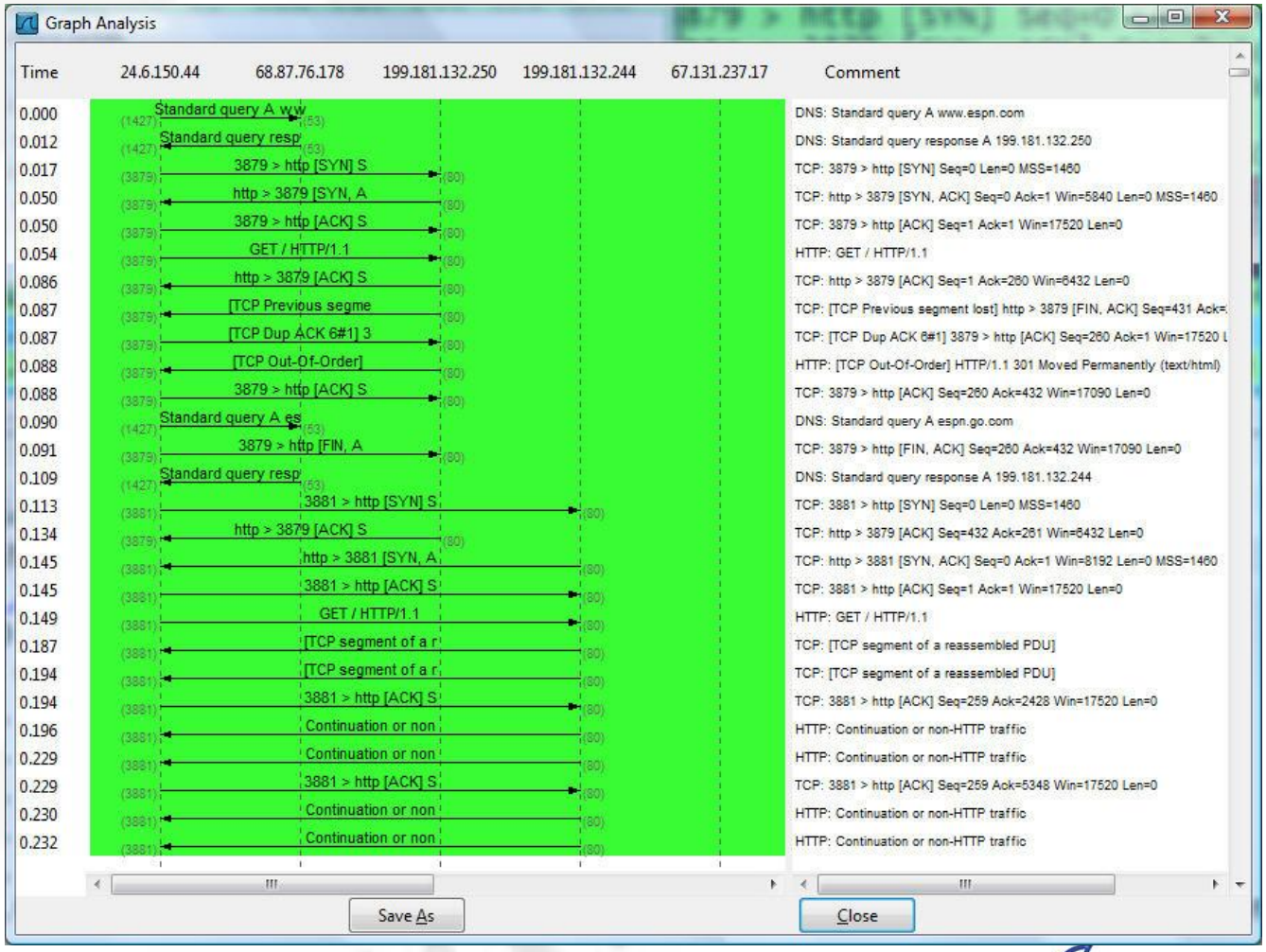
Analyze Unusual HTTP Traffic

http-espn.pcap

Notes:

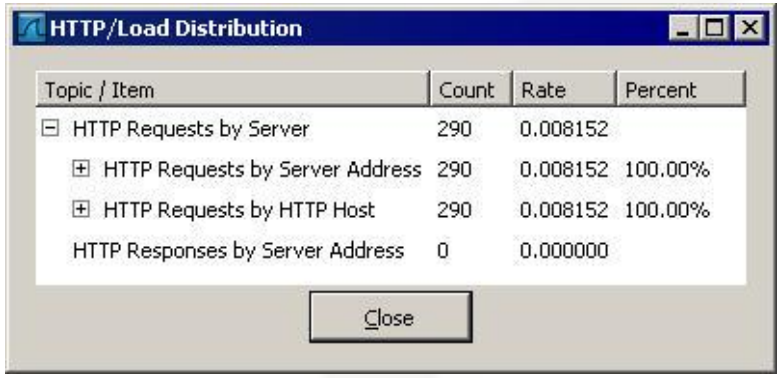
Dependencies on other web sites.

Not all HTTP requests are successful.

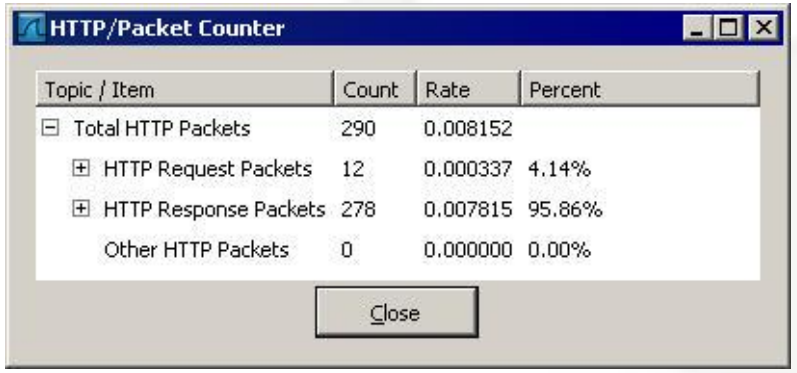


HTTP Statistics

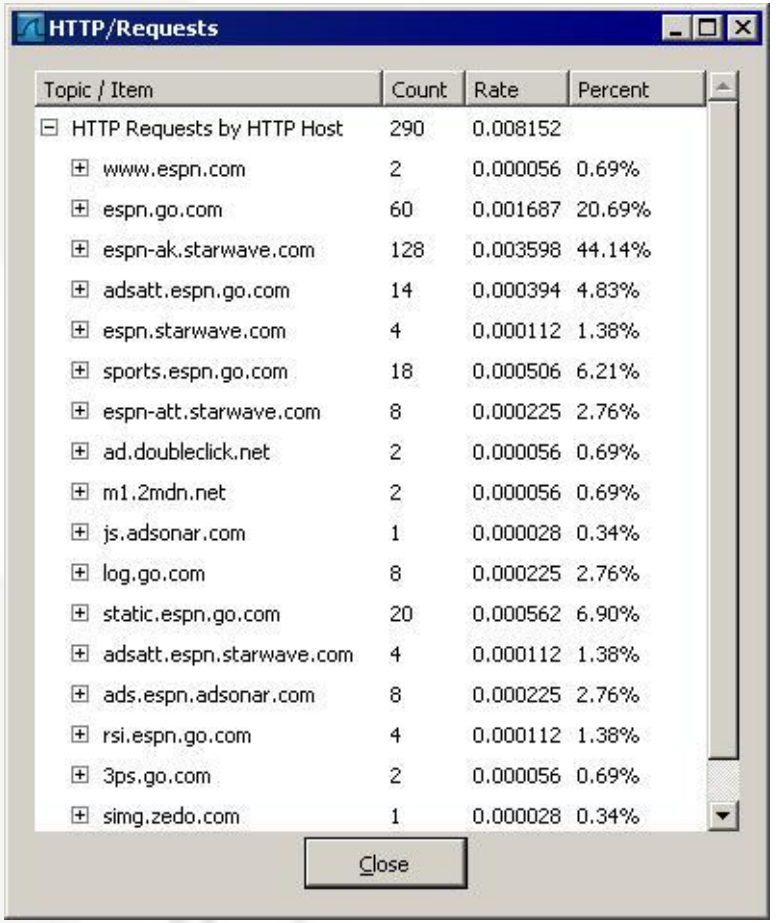
Load Distribution



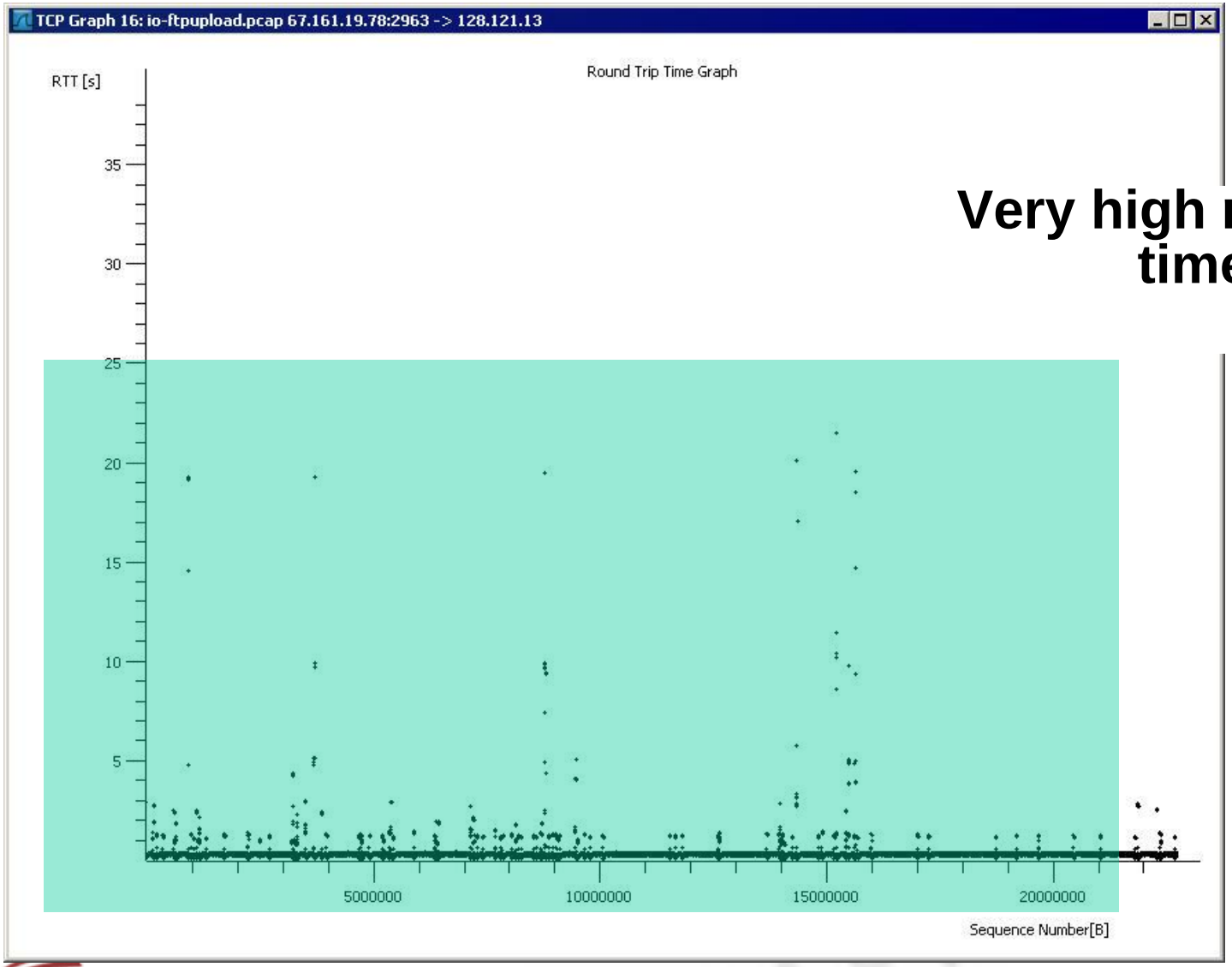
Packet Counter



Requests

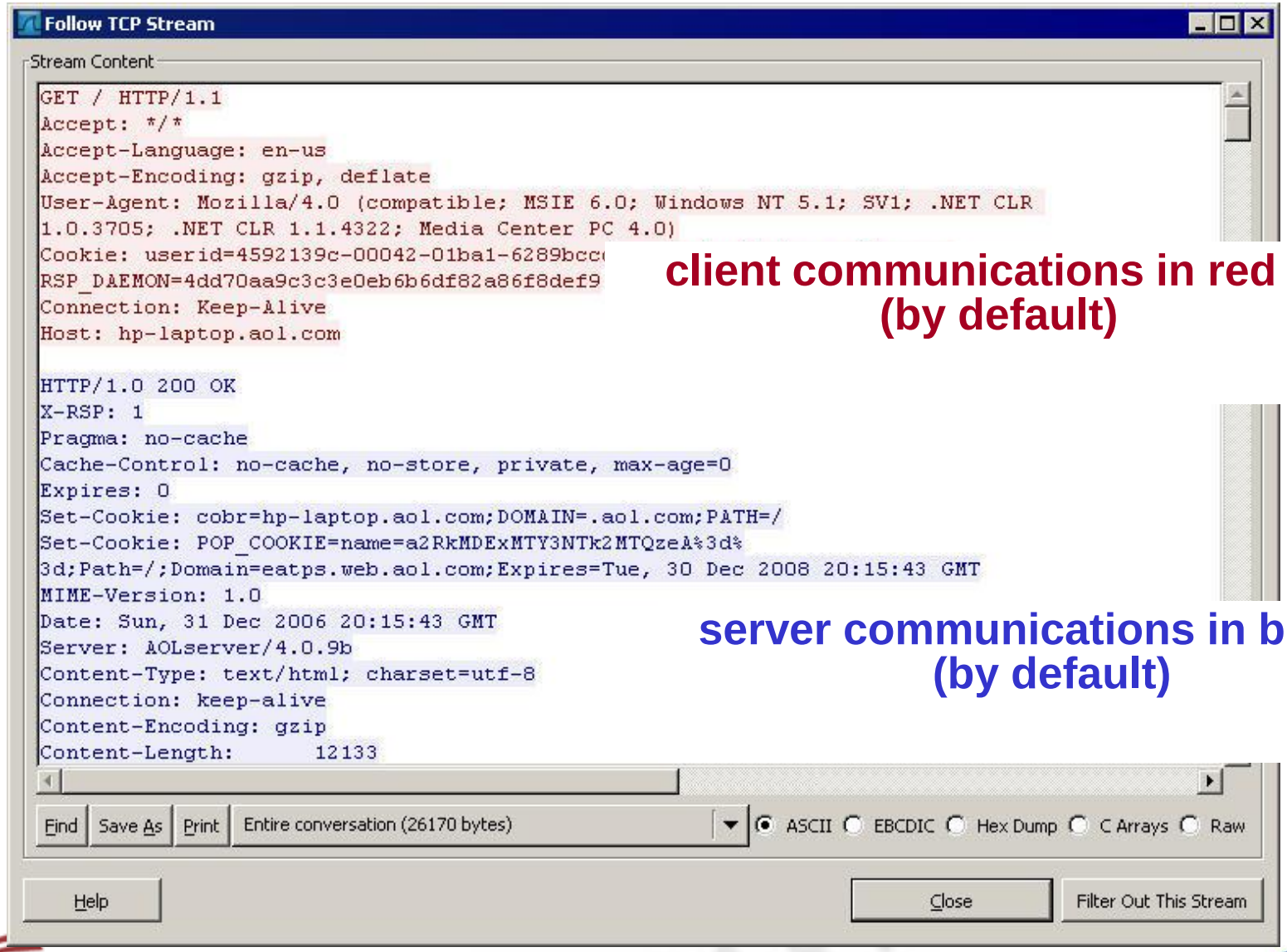


Round Trip Time Graphs



Very high round trip times!

TCP Stream Graphs

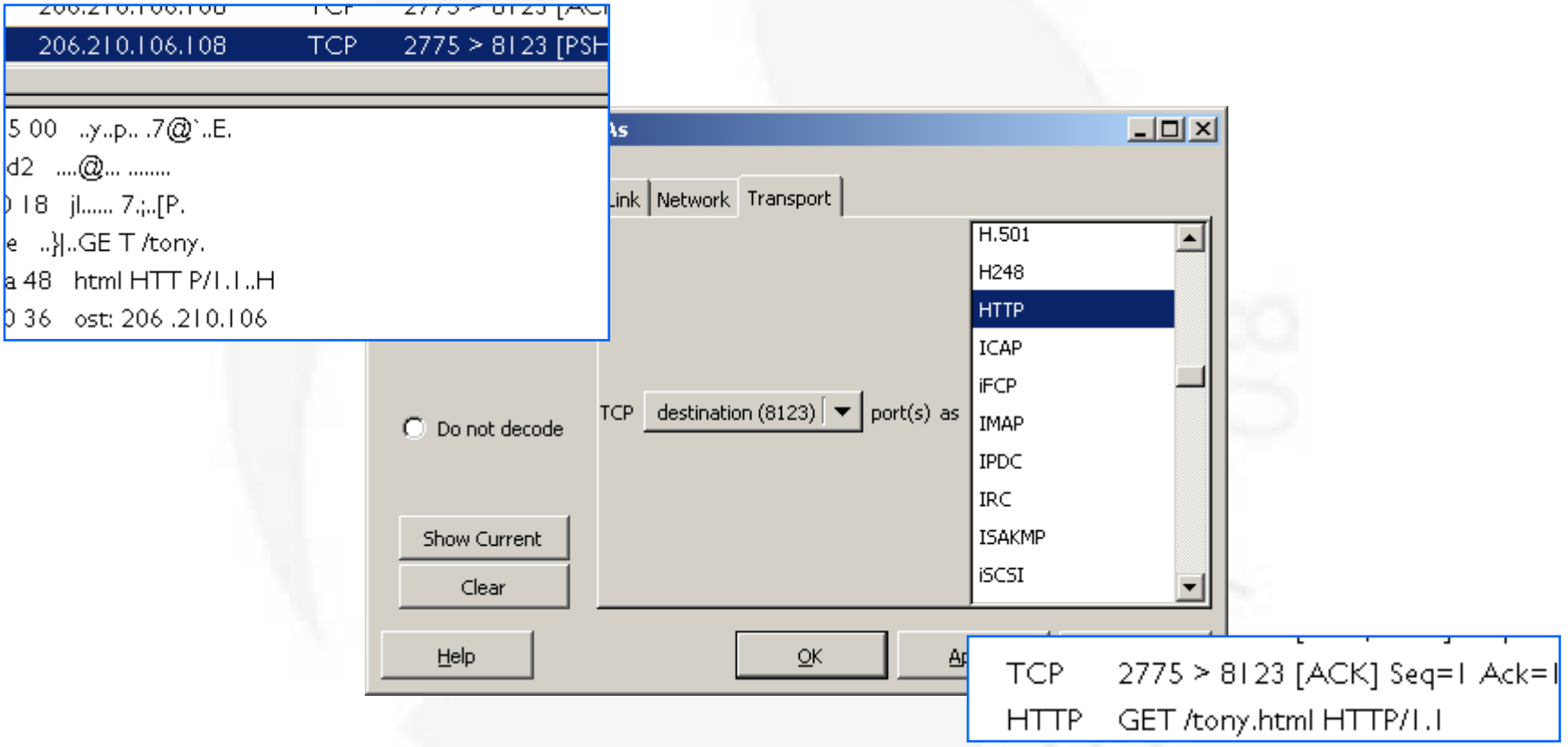


client communications in red (by default)

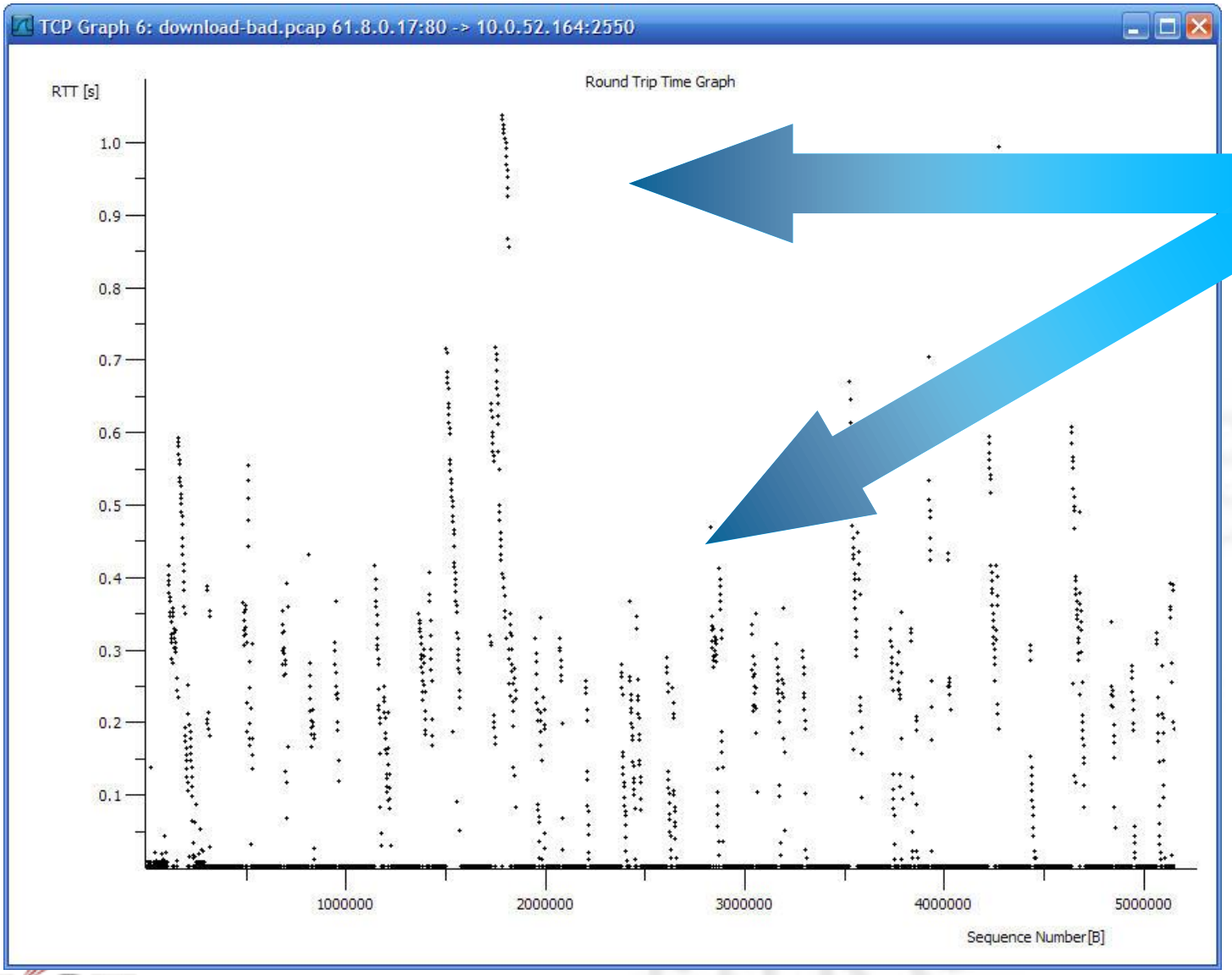
server communications in blue (by default)

Decode As

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

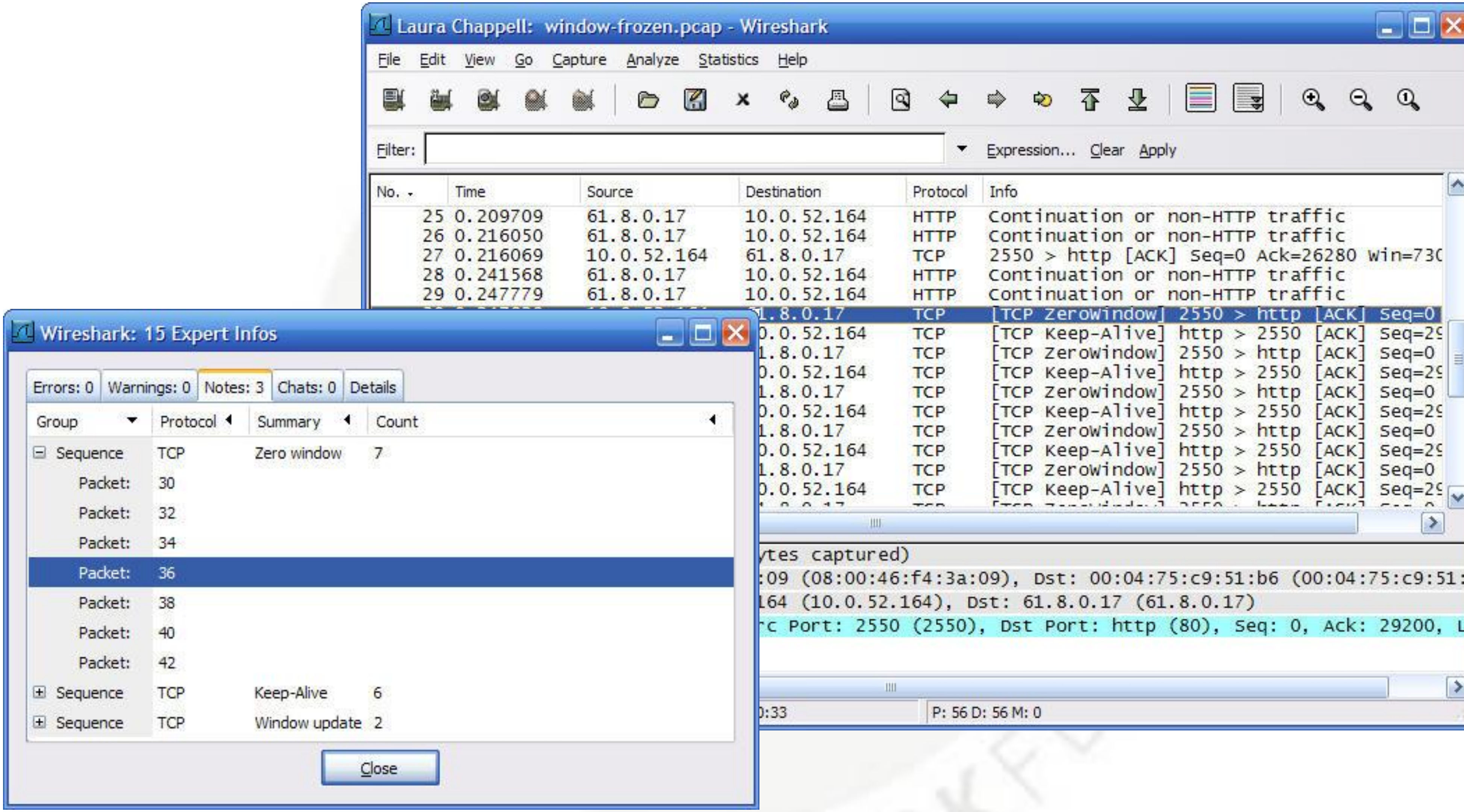


Using the Round Trip Time Graph



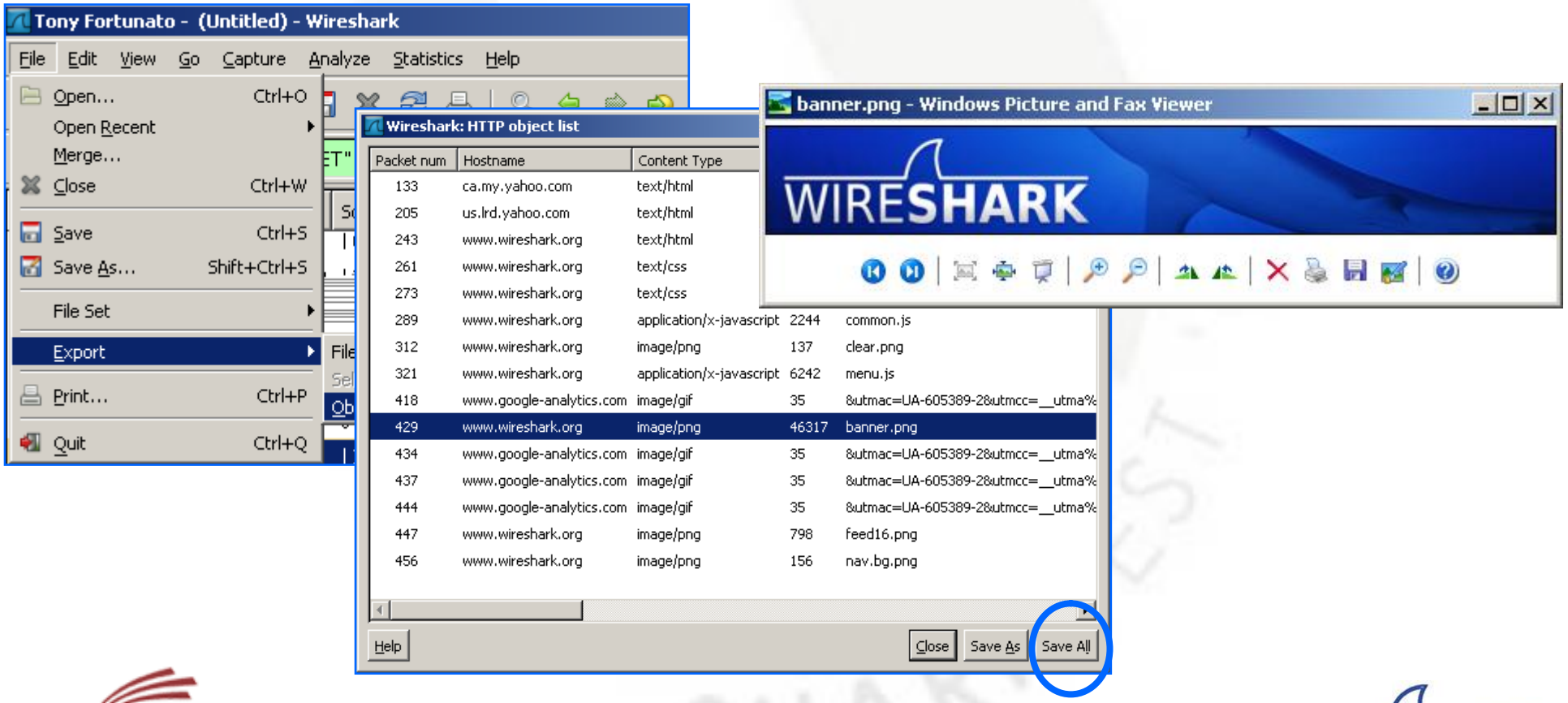
unacceptable latency times

TCP Receiver Congestion



File->Export->Objects->HTTP

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



CACE TECHNOLOGIES

www.wiresharkU.com WIRESHARK UNIVERSITY

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.