

Wireshark 圖形顯示分析網絡數據

Posted on 2019-03-13 Edited on 2022-06-08 In [Tools](#), [Network](#) Views: 759

Symbols count in article: 2k Reading time \approx 2 mins.

<https://breeztemple.github.io/2019/03/13/wireshark-statistics/>

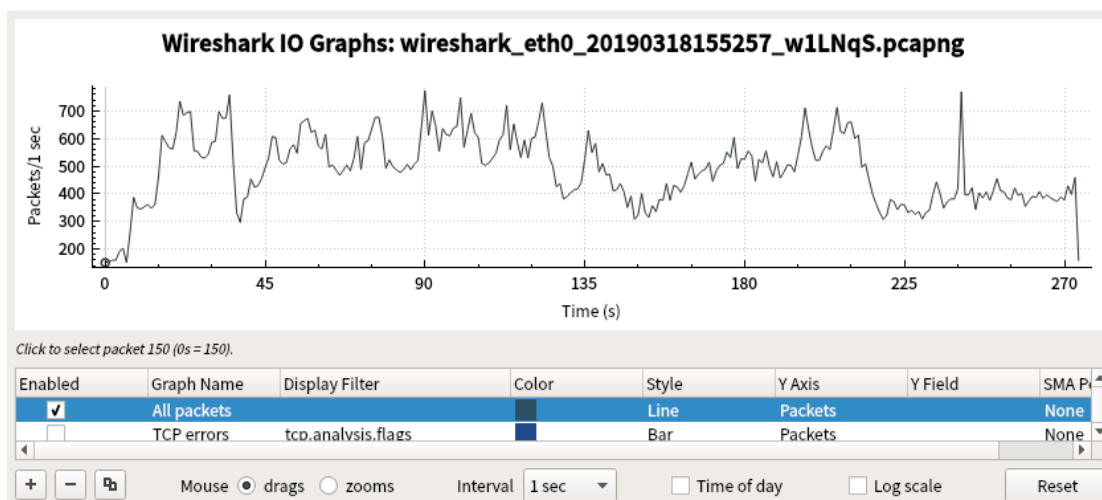
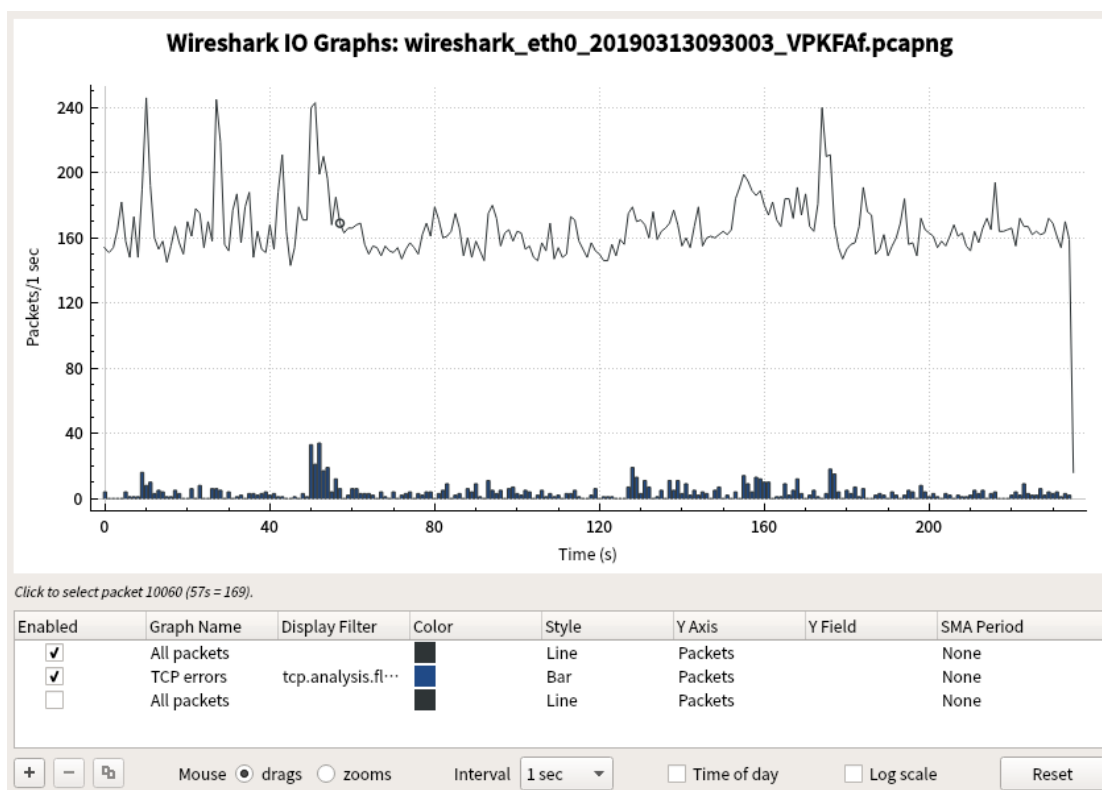
wireshark statistics 常用圖形工具

- IO Graph
- Flow Graph
- HTTP Packet Counter
- HTTP Requests
- HTTP Request Sequences
- TCP Throughput
- TCP Window Scaling
- TCP Round Trip Time

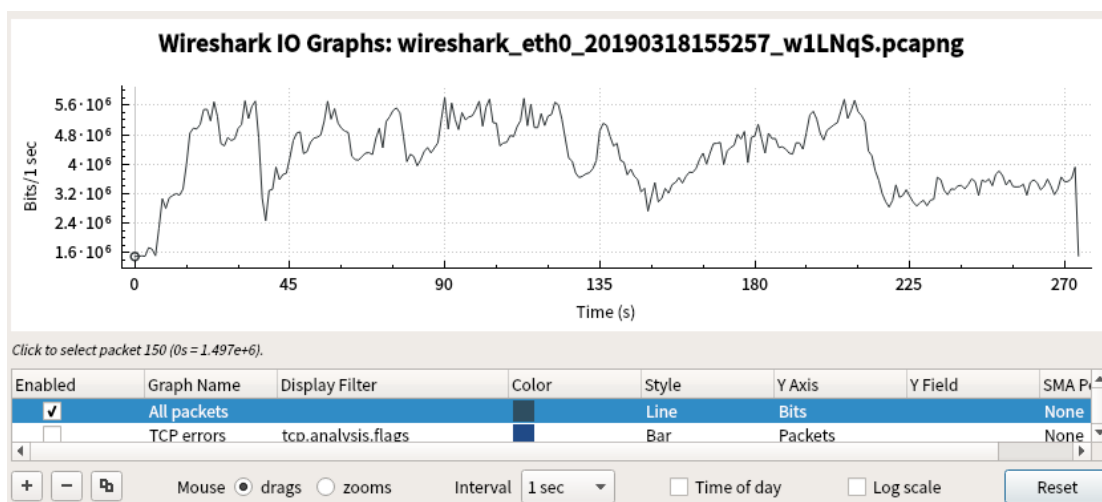
IO Graph

IO Graphs 這個窗口可以讓我們對網絡上的數據吞吐情況進行繪圖。這樣就可以很容易地發現數據吞吐的峰值，找出不同協議中的性能瓶頸，並且還可以用來比較實時的數據流。

選中任意一個 TCP 數據包，在菜單欄選擇“**Statistics**”->“**IO Graphs**”。這是一台電腦從互聯網下載文件時的例子：



過濾條件為空，此圖形顯示所有流量。雙擊 Y Axis 將 Y 軸改為 bits/tick 這樣就可以看到每秒的流量



常用過濾條件

1. `tcp.analysis.lost_segment`: 表明已經在抓包中看到不連續的序列號。報文丟失會造成重複的 ACK，這會導致重傳
2. `tcp.analysis.duplicate_ack`: 顯示被確認過不止一次的報文。大量的重複 ACK 是 TCP 端點之間高延時的跡象
3. `tcp.analysis.retransmission`: 顯示抓包中的所有重傳。如果重傳次數不多的話還是正常的，過多重傳可能有問題。這通常意味著應用性能緩慢和 / 或用戶報文丟失
4. `tcp.analysis.window_update`: 將傳輸過程中的 TCP window 大小圖形化。如果看到窗口大小下降為零，這意味著發送方已經退出了，並等待接收方確認所有已傳送數據。這可能表明接收端已經不堪重負了
5. `tcp.analysis.bytes_in_flight`: 某一時間點網絡上未確認字節數。未確認字節數不能超過你的 TCP 窗口大小（定義於最初 3 此 TCP 握手），為了最大化吞吐量你想要獲得盡可能接近 TCP 窗口大小。如果看到連續低於 TCP 窗口大小，可能意味著報文丟失或路徑上其他影響吞吐量的問題

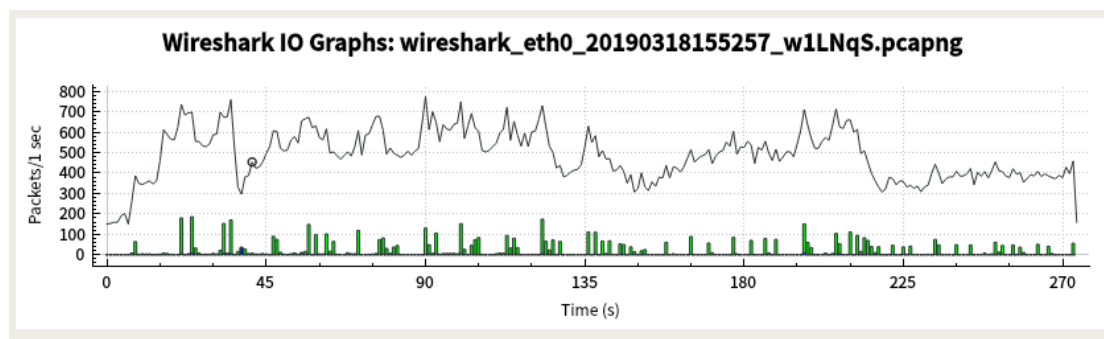
6. `tcp.analysis.ack_rtt`：衡量抓取的 TCP 報文與相應的 ACK。如果這一時間間隔比較長那可能表示某種類型的網絡延時（報文丟失，擁塞，等等）

在抓包中應用以上一些過濾條件：

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period
<input checked="" type="checkbox"/>	All packets		Black	Line	Packets		None
<input type="checkbox"/>	TCP errors	<code>tcp.analysis.flags</code>	Yellow	Bar	Packets		None
<input checked="" type="checkbox"/>	All packets	<code>tcp.analysis.duplicate_ack</code>	Green	Bar	Packets		None
<input checked="" type="checkbox"/>	All packets	<code>tcp.analysis.lost_segment</code>	Red	Bar	Packets		None
<input checked="" type="checkbox"/>	All packets	<code>tcp.analysis.retransmission</code>	Blue	Bar	Packets		None

+ - [Icon] Mouse ☒ drags ☐ zooms Interval 1 sec [v] ☐ Time of day

得到如下圖：

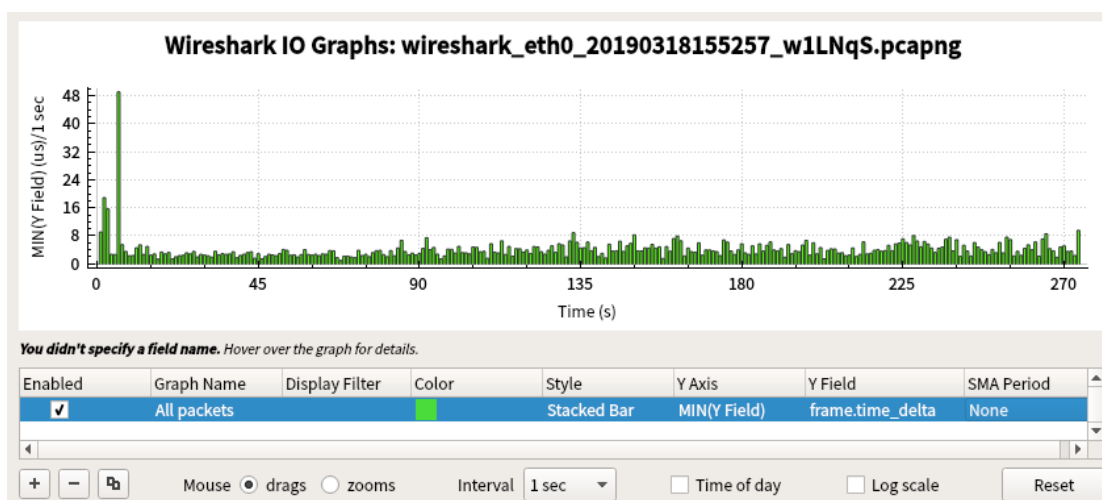


- Line 是 HTTP 總體流量，顯示形式為 packets/tick，時間間隔 1 秒
- Red 是 TCP 丟失報文片段
- Green 是 TCP 重複 ACK
- Blue 重傳

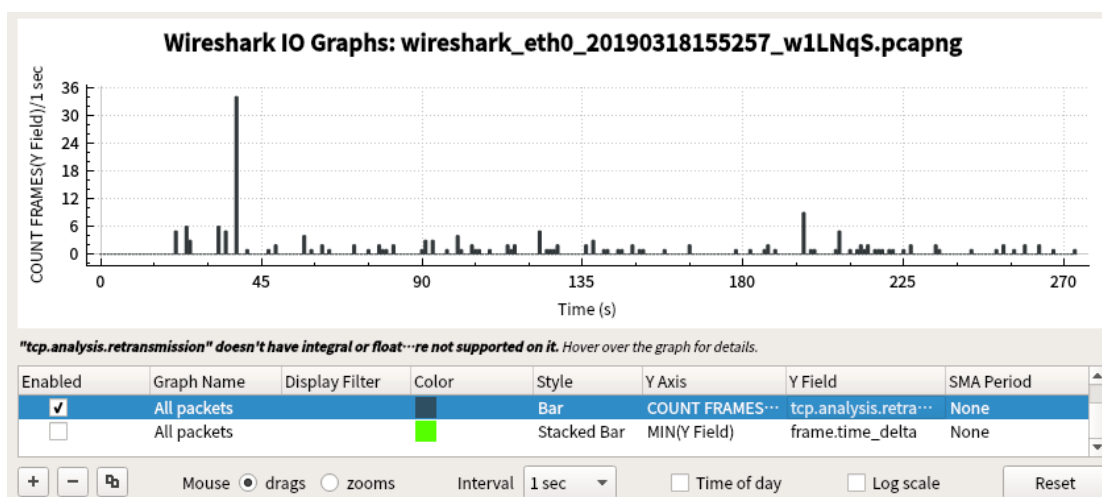
函數

IO Graphs 有六個可用函數：`SUM`, `MIN`, `AVG`, `MAX`, `COUNT`, `LOAD`

`MIN`、`AVG` 和 `MAX` 幀之間的最小，平均和最大時間，這對於查看幀 / 報文之間的延時非常有用可以將這些函數結合 `frame.time_delta` 過濾條件看清楚幀延時，並使得往返延時更為明顯。



COUNT 此函數計算時間間隔內事件發生的次數，在查看 TCP 分析標識符時很有用，例如重傳。例圖如下：



SUM 該函數統計事件的累加值。有兩種常見的用例是看在捕獲 TCP 數據量，以及檢查 TCP 序列號。

Flow Graph

數據流圖功能可以將連接可視化，並且將一段時間中的數據流顯示出來。數據流圖一般以列的方式將主機之間的連接顯示出來，並將數據組織到一起，便於更加直觀地解讀。

選擇菜單欄的 **Statistics->Flow Graph**，就可以打開數據流圖窗口：



從上圖中我們可以更好地看到整個連接的情況，比如 TCP 的三次握手，數據傳輸以及 HTTP 協議等的信息都一目瞭然。

Packet Counter

Statistics->HTTP->Packet Counter，每一個網站的報文數量。幫助識別有多少響應和請求

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Total HTTP Packets	863				0.0019	100%	0.0400	47.410
Other HTTP Packets	0				0.0000	0.00%	-	-
HTTP Response Packets	3				0.0000	0.35%	0.0100	5.530
??? : broken	0				0.0000	0.00%	-	-
5xx: Server Error	0				0.0000	0.00%	-	-
4xx: Client Error	0				0.0000	0.00%	-	-
3xx: Redirection	0				0.0000	0.00%	-	-
2xx: Success	3				0.0000	100.00%	0.0100	5.530
204 No Content	1				0.0000	33.33%	0.0100	215.596
200 OK	2				0.0000	66.67%	0.0100	5.530
1xx: Informational	0				0.0000	0.00%	-	-
HTTP Request Packets	860				0.0019	99.65%	0.0400	47.410
SEARCH	785				0.0017	91.28%	0.0300	46.409
NOTIFY	69				0.0001	8.02%	0.1300	327.035
GET	6				0.0000	0.70%	0.0300	436.829

Requests

Statistics->HTTP->Requests，各網站的請求分佈

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Bt
▼ HTTP Requests by HTTP Host	924				2.0396	100%	0.
▼ oosp.digicert.com	2				0.0044	0.22%	0.
/MHEwbzBNMEswSTAJBgUrDgMCGGUABBTfqhLjKLEJQZPin0KCzkdA...	1				0.0022	50.00%	0.
/MHEwbzBNMEswSTAJBgUrDgMCGGUABBRJ9L2KGL92BpjF3kAtaDtx...	1				0.0022	50.00%	0.
▼ connectivity-check.ubuntu.com	1				0.0022	0.11%	0.
/	1				0.0022	100.00%	0.
▼ [FF02::C]:1900	169				0.3730	18.29%	0.
*	169				0.3730	100.00%	0.
▼ 239.255.255.250:1900	748				1.6511	80.95%	0.
*	748				1.6511	100.00%	0.
▼ 192.168.110.254	4				0.0088	0.43%	0.
/redmine/issues/137354	1				0.0022	25.00%	0.
/gerrit/changes/?q=change:47475+has:draft&O=0	1				0.0022	25.00%	0.
/gerrit/changes/47475/edit?list	1				0.0022	25.00%	0.
/gerrit/changes/47475/detail?O=10004	1				0.0022	25.00%	0.

Display filter:

Request Sequences

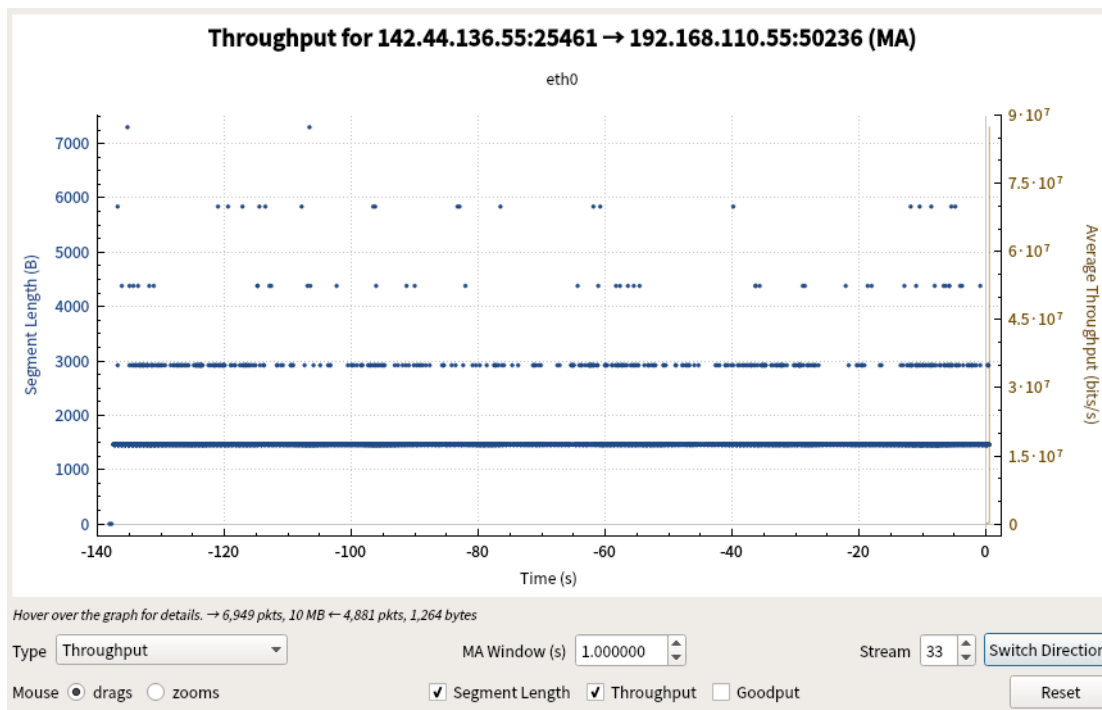
Statistics->HTTP->Request Sequences

Topic / Item	Cou	Average	Min val
▼ HTTP Request Sequences	76		
▼ http://192.168.110.254/redmine/issues/140569	2		
http://192.168.110.254/redmine/issues/137354	1		
▼ http://192.168.110.254/gerrit/	6		
http://192.168.110.254/gerrit/changes/47475/detail?O=10004	1		
http://192.168.110.254/gerrit/changes/47475/edit?list	1		
http://192.168.110.254/gerrit/changes/?q=change:47475+has:draft&O=0	1		
▼ http://[FF02::C]:1900*	36		
http://[fe80::6859:9b52:b987:7ca1]:2869/upnphost/udhisapi.dll?content=uuid:2b9f1b67-7500-4464-a...	18		
▼ http://239.255.255.250:1900*	108		
http://192.168.111.2:1327/	6		
http://192.168.110.253:8200/rootDesc.xml	12		
http://192.168.111.18:2869/upnphost/udhisapi.dll?content=uuid:2b9f1b67-7500-4464-ad0a-1151f02...	18		
http://192.168.111.253:5000/ssdp/desc-DSM-eth0.xml	18		

Display filter:

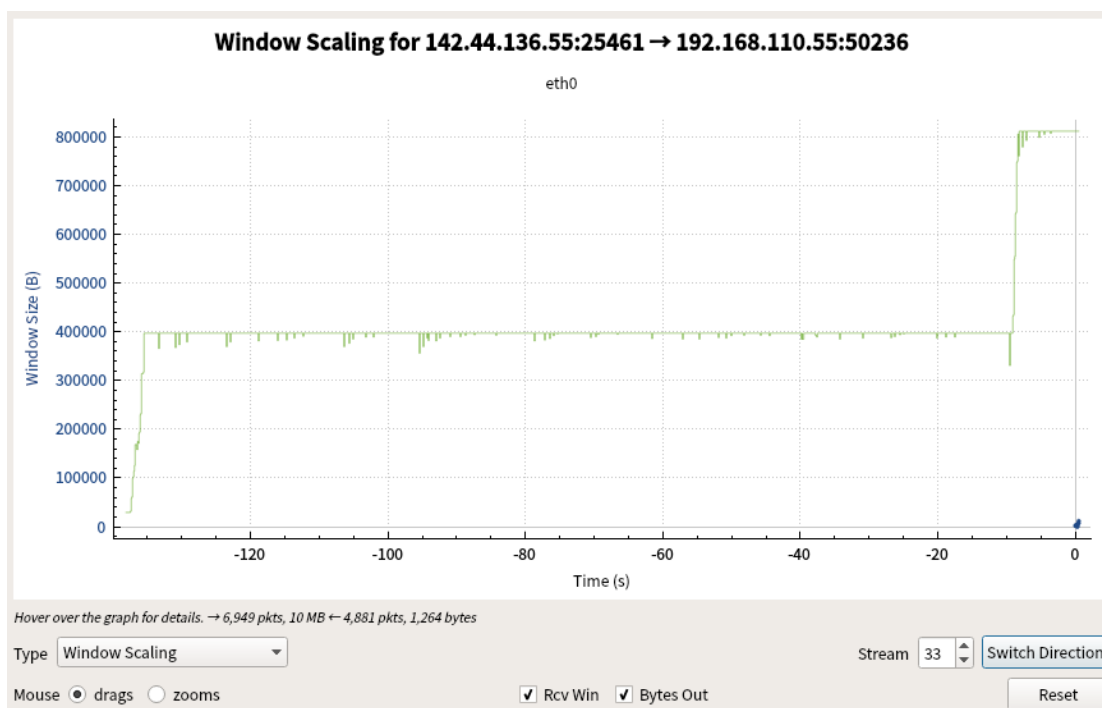
Throughput

Statistics->TCP->Throughput，顯示 TCP 流吞吐量圖形



Window Scaling

顯示 TCP 滑動窗口圖形



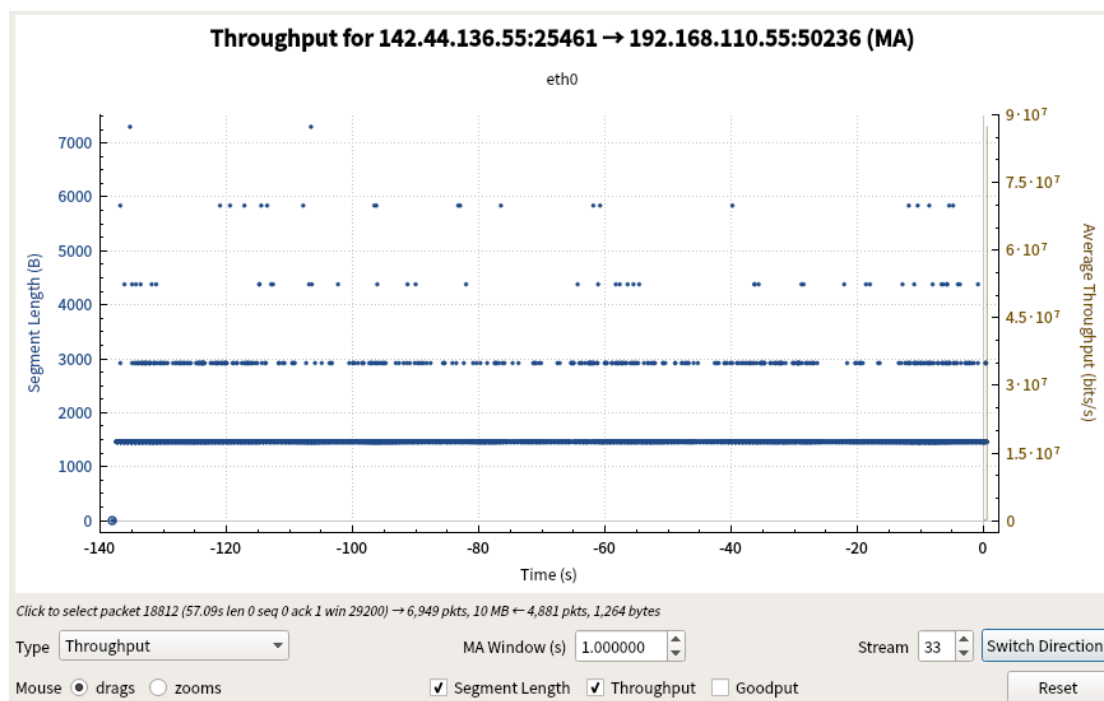
Round Trip Time

Wireshark 的另一個繪圖功能就是對所捕獲的文件進行往返時間的繪圖。往返時間

（round-trip time, RTT）是指一個數據包從發出到確認被成功接收所需要的時間。

或者說，往返時間就是數據包抵達目的地的時間，加上收到對方的確認信息的時間之和。通過對這個時間的分析，可以找到通信中的瓶頸，確定是否存在延遲。

選擇 **Statistics->TCP Stream Graph->Round Trip Time Graph**，來查看往返時間圖：



上圖中的每個點代表的是一個數據包的往返時間。在默認情況下，這些值按照序號進行排序。單擊圖中任意一個點，就可以在 **Packet List** 面板中看到相應的數據包。