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 ≈ 2 mins.

https://breezetemple.github.io/2019/03/13/wireshark-statistics/

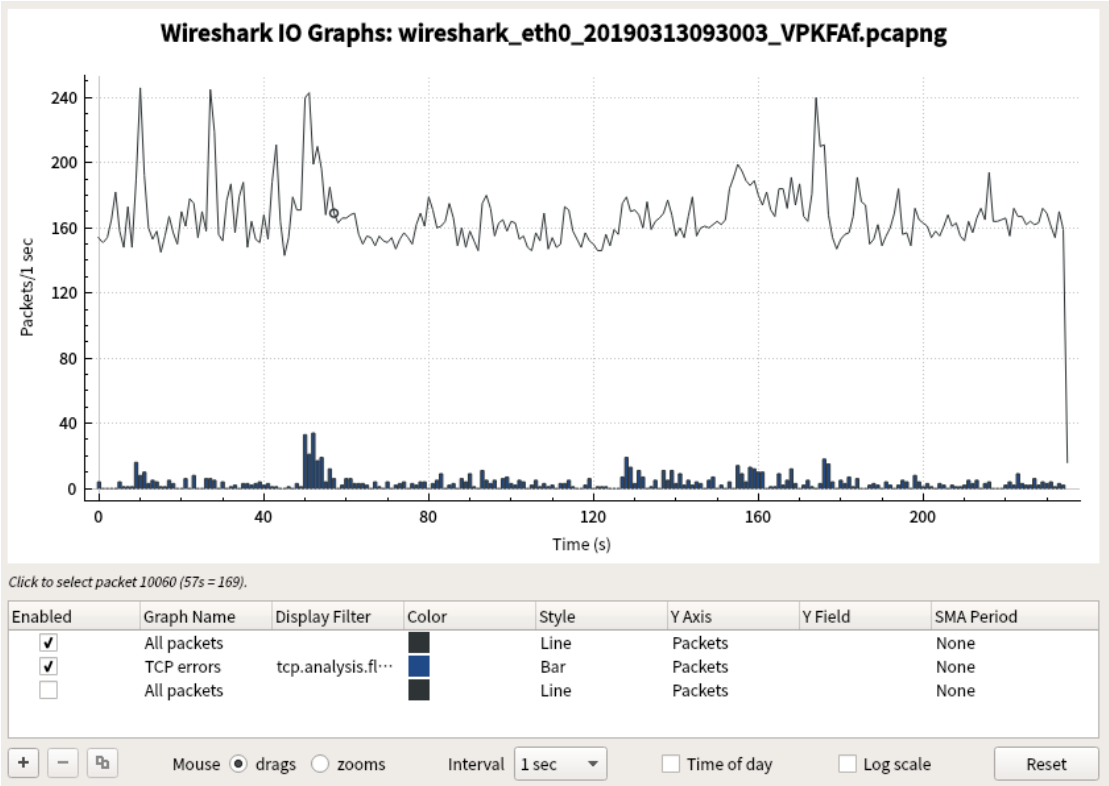
`wireshark statistics`常用圖形工具

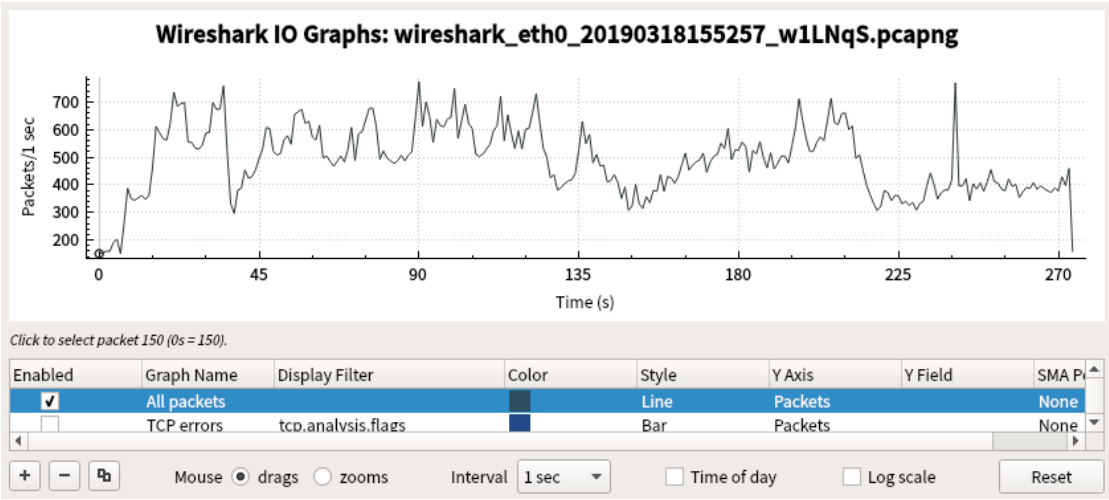
- IO Graph
- Flow Graph
- HTTP Packet Counter
- HTTP Requests
- HTTP Request Sequences
- TCP Throughput
- TCP Window Scaling
- TCP Round Trip Time

IO Graph

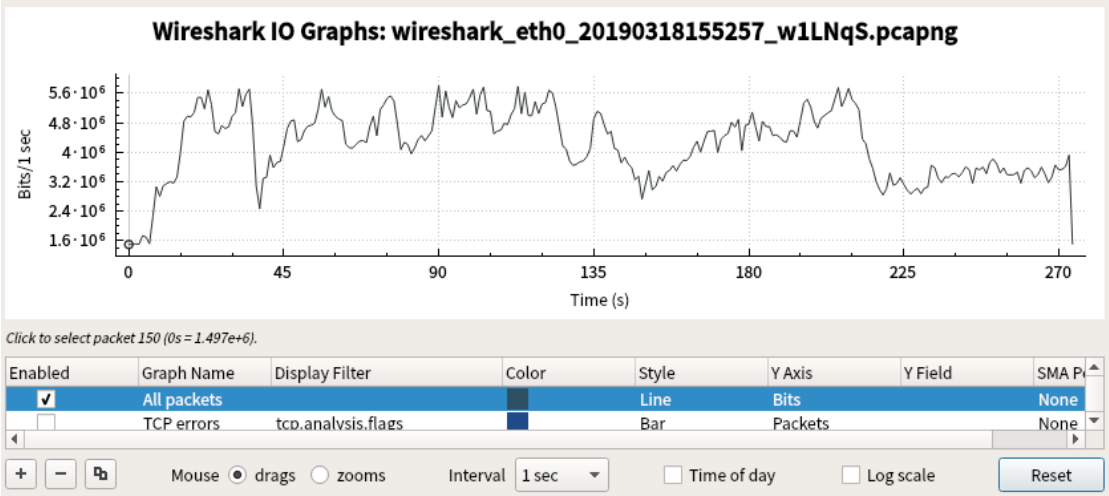
`IO Graphs`這個窗口可以讓我們對網絡上的數據吞吐情況進行繪圖。這樣就可以很容易地發現數據吞吐的峰值，找出不同協議中的性能瓶頸，並且還可以用來比較實時的數據流。

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**

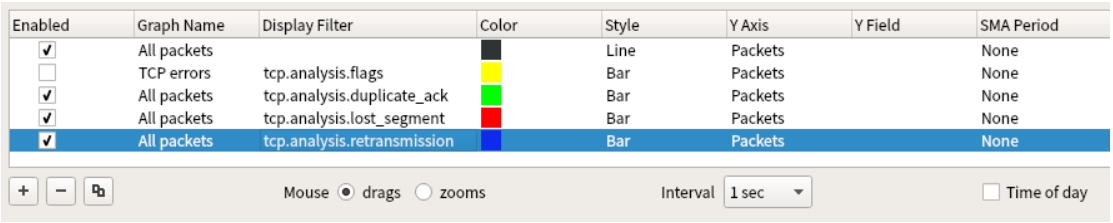


常用過濾條件

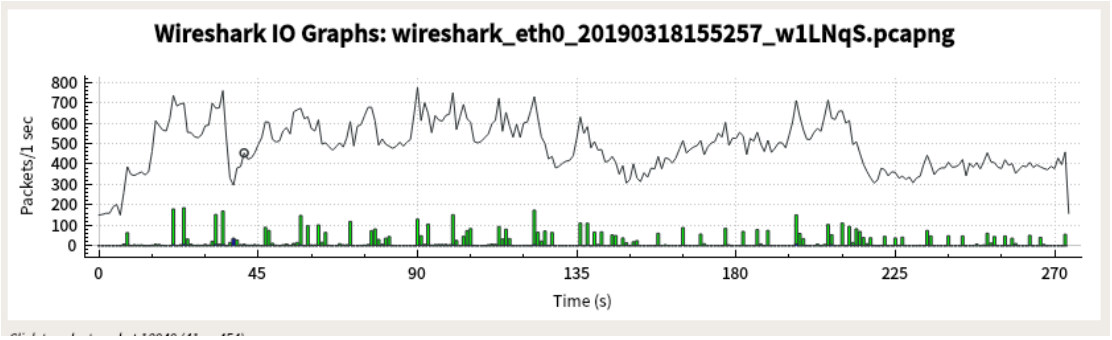
1. `tcp.analysis.lost_segment`：表明已經在抓包中看到不連續的序列號。報文丟失會造成重復的 ACK，這會導致重傳
2. `tcp.analysis.duplicate_ack`：顯示被確認過不止一次的報文。大量的重復 ACK 是 TCP 端點之間高延時的跡象
3. `tcp.analysis.retransmission`：顯示抓包中的所有重傳。如果重傳次數不多的話還是正常的，過多重傳可能有問題。這通常意味著應用性能緩慢和 / 或用戶報文丟失
4. `tcp.analysis.window_update`：將傳輸過程中的 TCP window 大小圖形化。如果看到窗口大小下降為零，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：



得到如下圖：

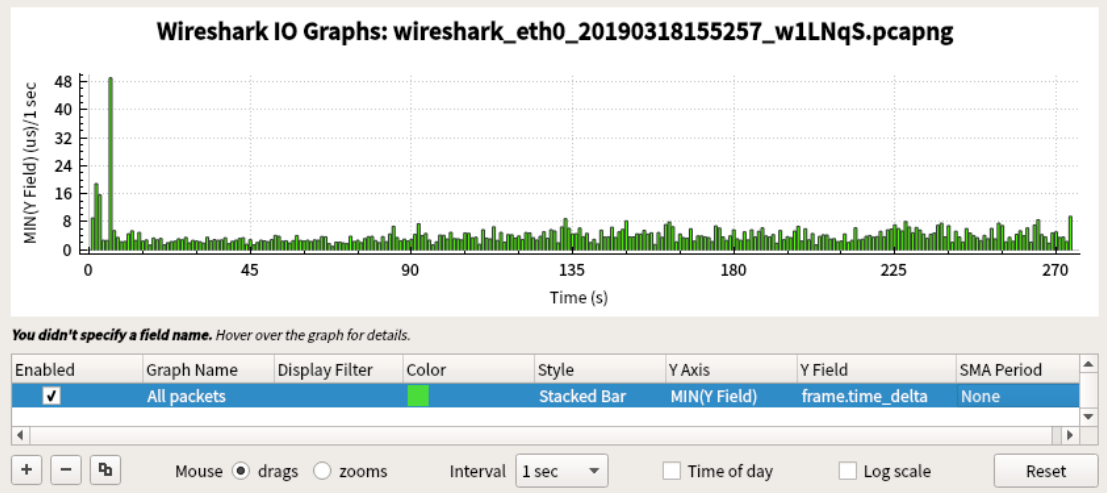


- Line 是 HTTP 總體流量，顯示形式為 packets/tick，時間間隔 1 秒
- Red 是 TCP 丟失報文片段
- Green 是 TCP 重復 ACK
- Blue 重傳

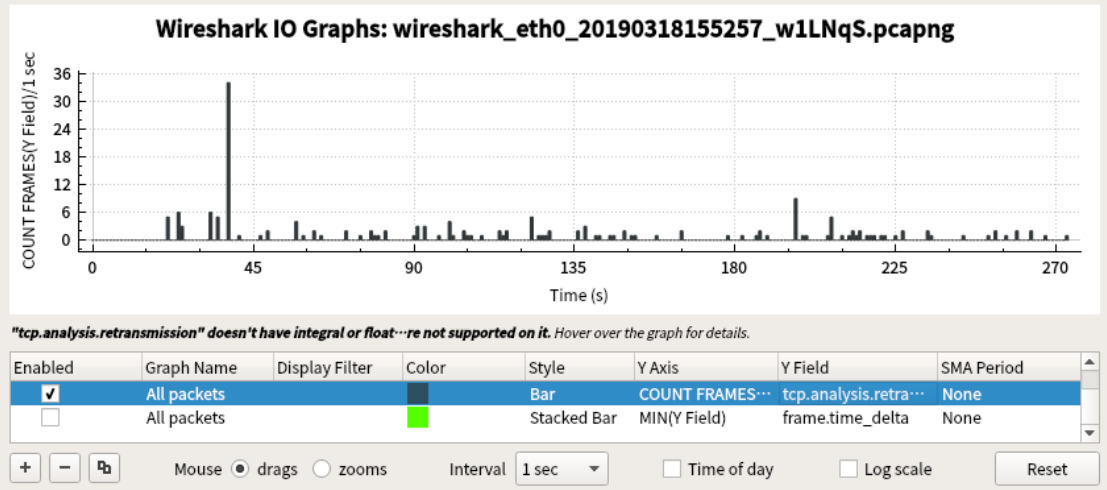
函數

`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：

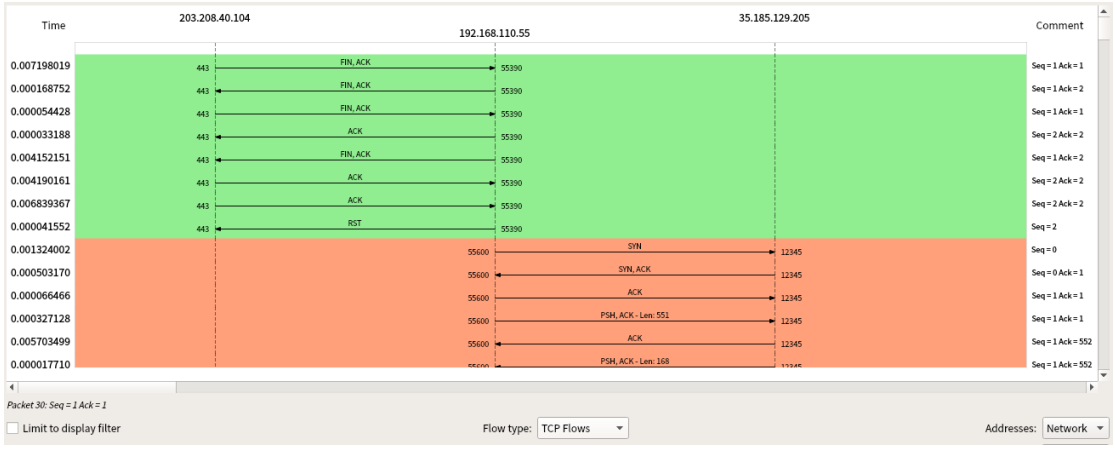


SUM 該函數統計事件的累加值。有兩種常見的用例是看在捕獲 TCP 數據量，以及檢查 TCP 序列號。

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。

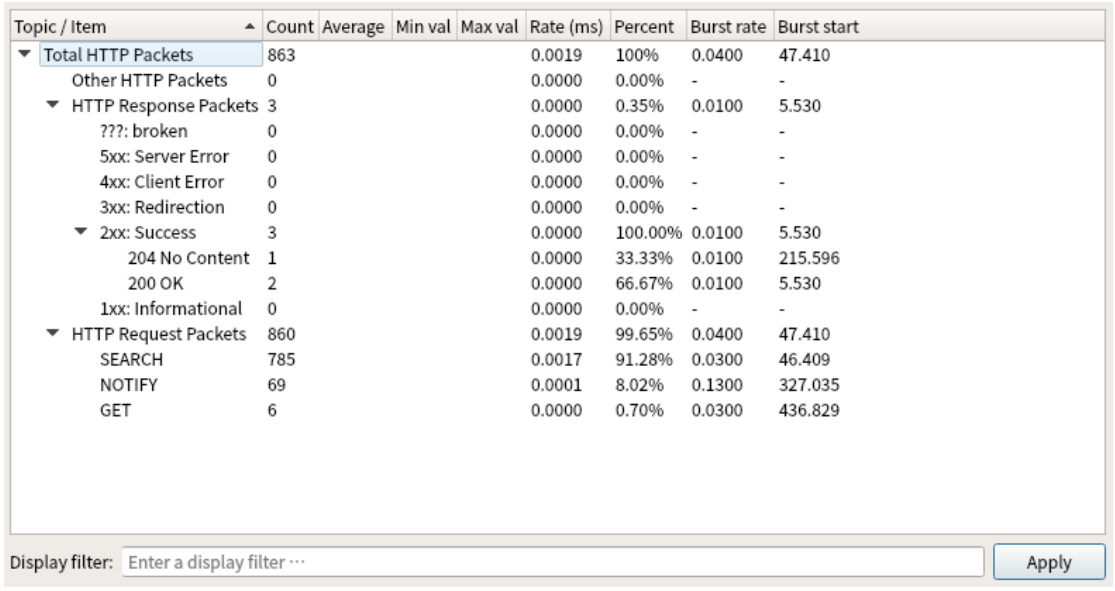
選擇菜單欄的 **Statistics->Flow Graph**，就可以打開數據流圖窗口：



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。

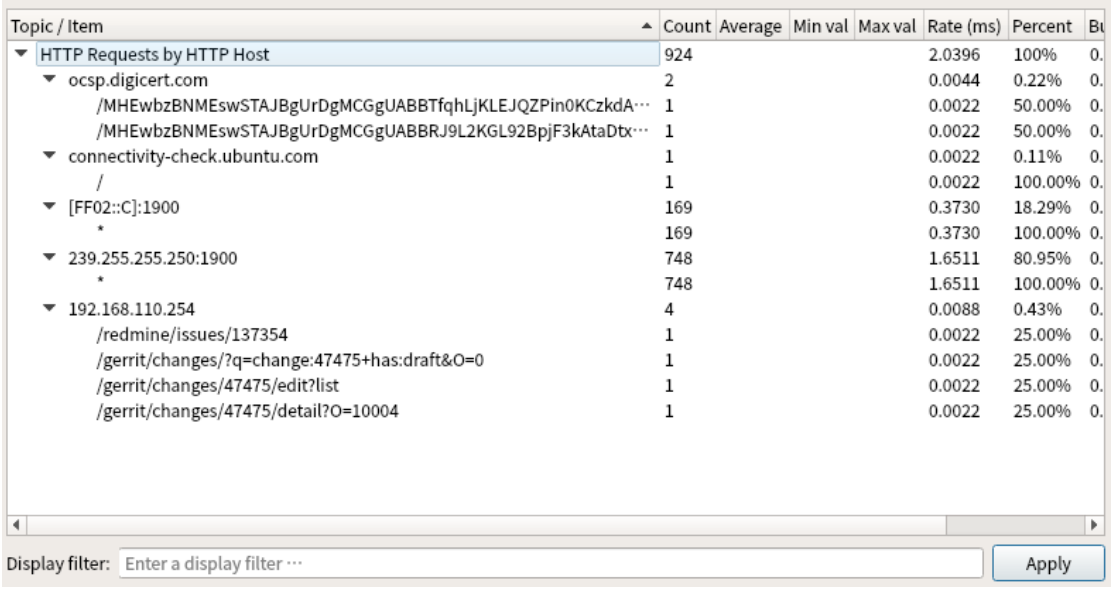
Packet Counter

`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



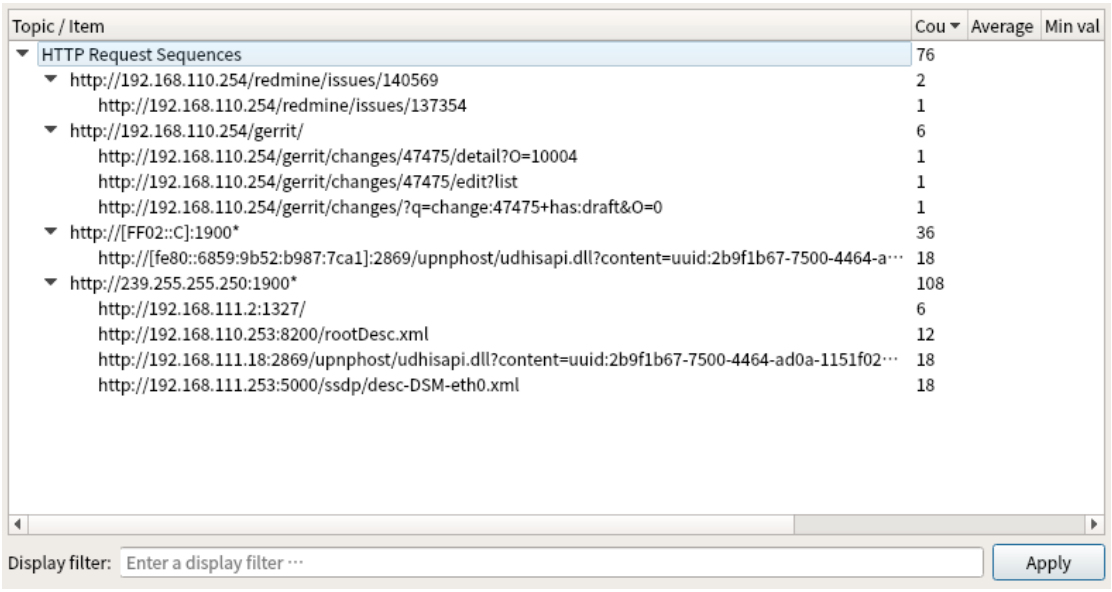
Requests

`Statistics->HTTP->Requests`，各網站的請求分佈



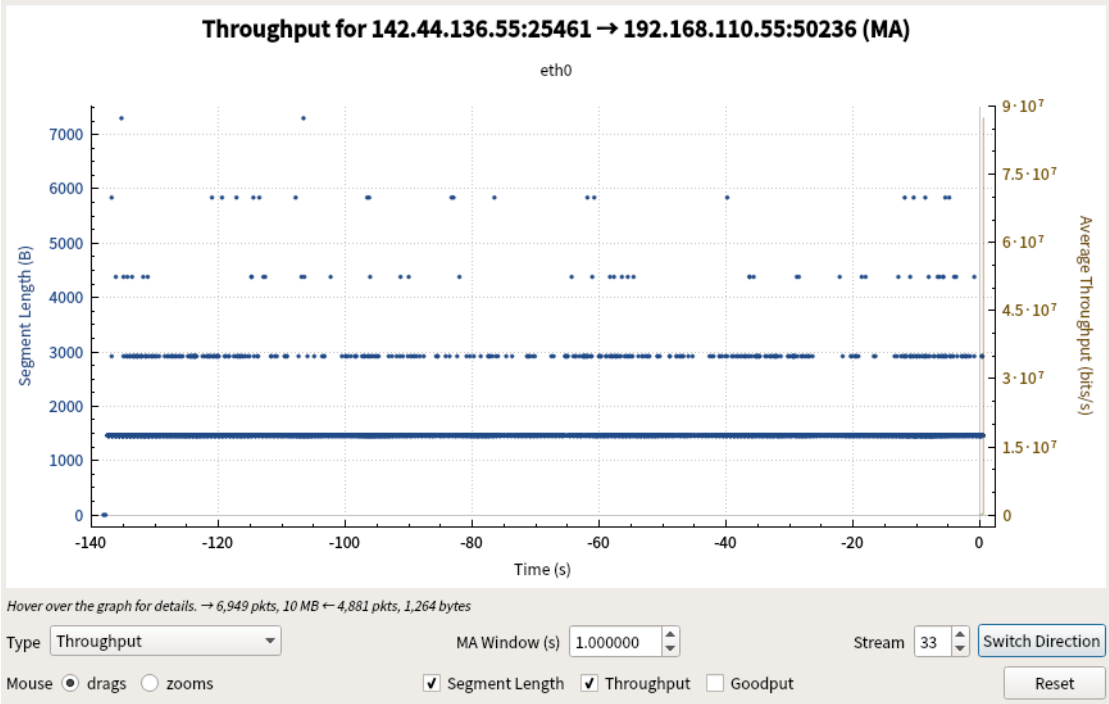
Request Sequences

`Statistics->HTTP->Request Sequences`



Throughput

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



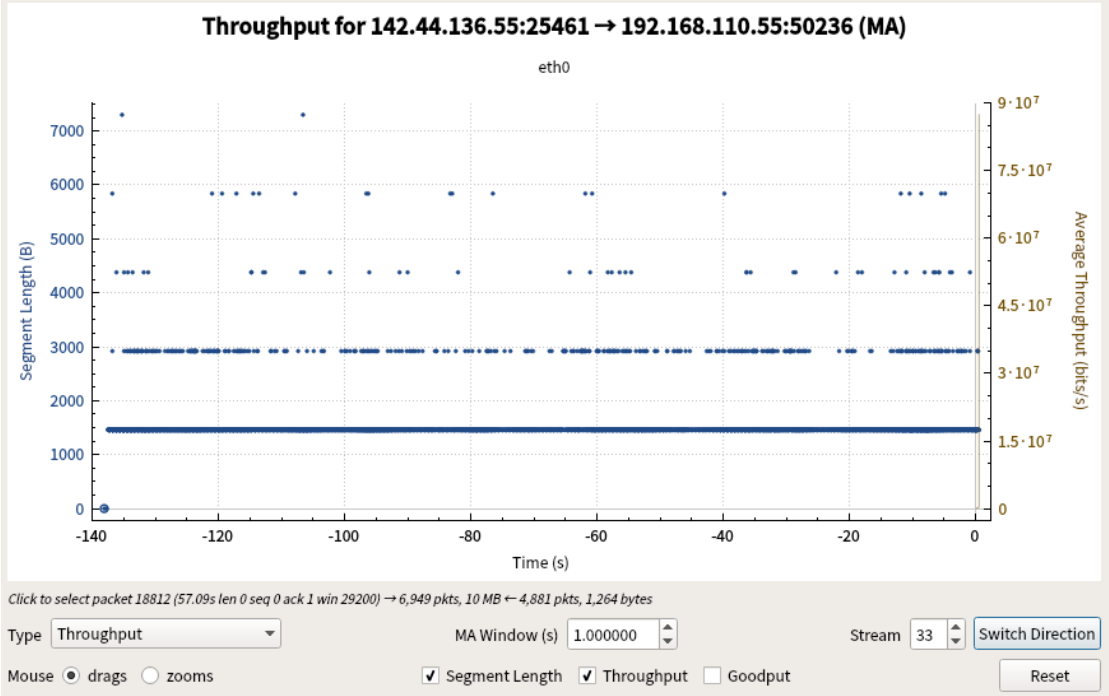
Window Scaling

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 Stram 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。