ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Computer Networks: Lab 6 - TCP

Luca Bedogni

Department of Computer Science and Engineering
University of Bologna

A brief introduction

- We use Wireshark to obtain a packet trace of the TCP transfer of a file from your computer to a remote server.
  - We will access a Web page through which we will send a file in our computer to a server

- At first download this: http://gaia.cs.umass.edu/wireshark-labs/alice.txt
  - This is the file we will transfer

- We will use the POST method since it a rather large file

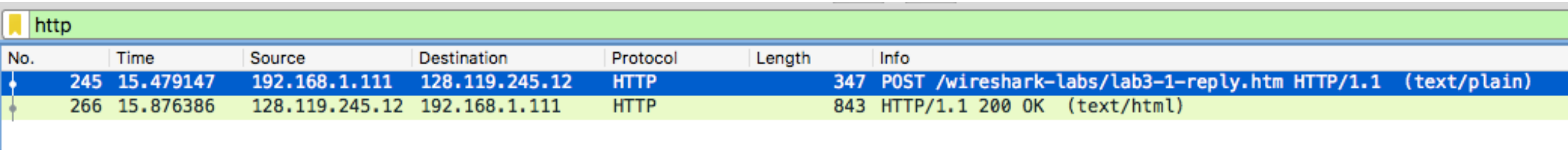
TCP Lab

- After you've downloaded alice.txt, go to http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html
- Start wireshark packet capture
- Browse the file and select alice.txt
- Upload it
- Stop wireshark packet capture
- Or use *tcp-ethereal-trace-1*

- Filter out by writing **tcp** as display filter
  - Why are you seeing also HTTP packets?

- Let's find the HTTP post message

TCP Lab



- You can see 1 POST request

TCP Lab

- What are the Source and Destination IP/Port

248	15.502748	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=106380 Win=183296 Len=0 TSval=1359918649 TSecr=1059143622
249	15.526630	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=109276 Win=183296 Len=0 TSval=1359918673 TSecr=1059143643
250	15.549134	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=112172 Win=183296 Len=0 TSval=1359918696 TSecr=1059143665
251	15.572506	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=115068 Win=183296 Len=0 TSval=1359918719 TSecr=1059143686
252	15.595519	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=117964 Win=183296 Len=0 TSval=1359918742 TSecr=1059143697
253	15.617994	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=120860 Win=183296 Len=0 TSval=1359918765 TSecr=1059143708
254	15.641842	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=123756 Win=183296 Len=0 TSval=1359918788 TSecr=1059143718
255	15.664834	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=126652 Win=183296 Len=0 TSval=1359918811 TSecr=1059143730
256	15.688236	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=129548 Win=183296 Len=0 TSval=1359918834 TSecr=1059143741
257	15.710945	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=132444 Win=183296 Len=0 TSval=1359918857 TSecr=1059143752
258	15.740433	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=135340 Win=183296 Len=0 TSval=1359918881 TSecr=1059143763
259	15.757657	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=138236 Win=183296 Len=0 TSval=1359918904 TSecr=1059143773
260	15.780840	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=141132 Win=183296 Len=0 TSval=1359918927 TSecr=1059143784
261	15.802788	128.119.245.12	192.168.1.111	TCP	66	80 → 52248 [ACK] Seq=1 Ack=144028 Win=183296 Len=0 TSval=1359918949 TSecr=1059143795

```
▶ Frame 252: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▶ Ethernet II, Src: Technico_d7:50:66 (e0:b9:e5:d7:50:66), Dst: Apple_90:b9:76 (d0:e1:40:90:b9:76)
▶ Internet Protocol Version 4, Src: 128.119.245.12, Dst: 192.168.1.111
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 52248, Seq: 1, Ack: 117964, Len: 0
    Source Port: 80
    Destination Port: 52248
    [Stream index: 10]
    [TCP Segment Len: 0]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 117964    (relative ack number)
    1000 .... = Header Length: 32 bytes (8)
  ▶ Flags: 0x010 (ACK)
    Window size value: 1432
    [Calculated window size: 183296]
    [Window size scaling factor: 128]
    Checksum: 0x375f [unverified]
    [Checksum Status: Unverified]
```

Take a moment to identify SYN/ACK/FIN segments

TCP Lab – further questions

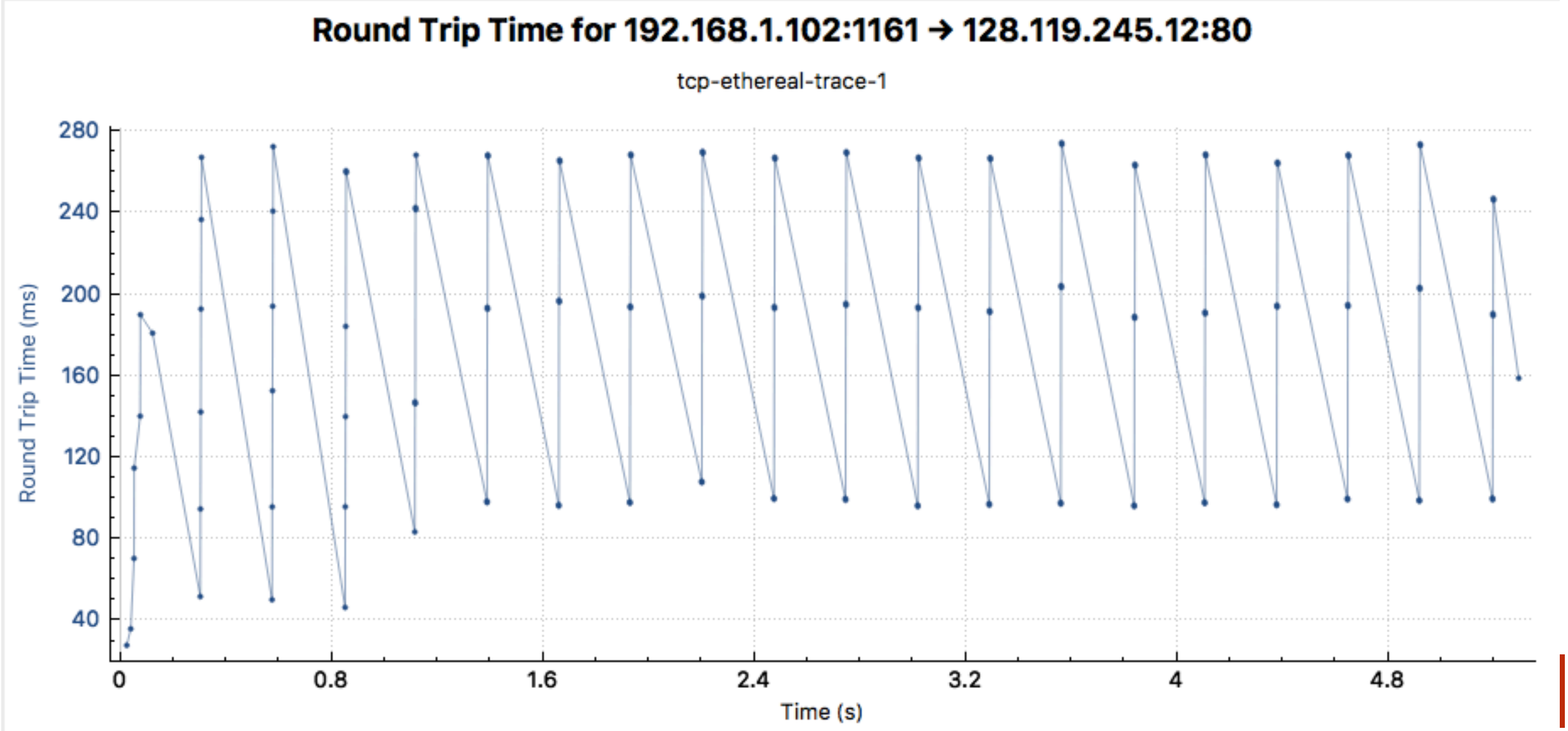
- How many ACKs are sent?
- Investigate the sequence numbers
- Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection.
  - What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)?
  - At what time was each segment sent? When was the ACK for each segment received?
  - Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments
- Let's now plot the RTT
  - Select a TCP Segment
  - Then select: *Statistics->TCP Stream Graph- >Round Trip Time Graph.*

TCP Lab – further questions

- How many ACKs are sent? Investigate the sequence numbers

```
  189 5.106121    192.168.1.100 192.168.1.1   SSDP      175 M-SEARCH * HTTP/1.1
  190 5.125019    128.119.245.12 192.168.1.102 TCP        60 80 → 1161 [ACK] Seq=1 Ack=154117 Win=62780 Len=0
  191 5.197286    128.119.245.12 192.168.1.102 TCP        60 80 → 1161 [ACK] Seq=1 Ack=156469 Win=62780 Len=0
  192 5.197508    192.168.1.102 128.119.245.12 TCP      1514 1161 → 80 [ACK] Seq=156469 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
  193 5.198388    192.168.1.102 128.119.245.12 TCP      1514 1161 → 80 [ACK] Seq=157929 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
  194 5.199275    192.168.1.102 128.119.245.12 TCP      1514 1161 → 80 [ACK] Seq=159389 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
  195 5.200252    192.168.1.102 128.119.245.12 TCP      1514 1161 → 80 [ACK] Seq=160849 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
  196 5.201150    192.168.1.102 128.119.245.12 TCP      1514 1161 → 80 [ACK] Seq=162309 Ack=1 Win=17520 Len=1460 [TCP segment of a reassembled PDU]
  197 5.202024    192.168.1.102 128.119.245.12 TCP       326 1161 → 80 [PSH, ACK] Seq=163769 Ack=1 Win=17520 Len=272 [TCP segment of a reassembled PDU]
  198 5.297257    128.119.245.12 192.168.1.102 TCP        60 80 → 1161 [ACK] Seq=1 Ack=159389 Win=62780 Len=0
  199 5.297341    192.168.1.102 128.119.245.12 HTTP      104 POST /ethereal-labs/lab3-1-reply.htm HTTP/1.1  (text/plain)
  200 5.389471    128.119.245.12 192.168.1.102 TCP        60 80 → 1161 [ACK] Seq=1 Ack=162309 Win=62780 Len=0
  201 5.447887    128.119.245.12 192.168.1.102 TCP        60 80 → 1161 [ACK] Seq=1 Ack=164041 Win=62780 Len=0
  202 5.455830    128.119.245.12 192.168.1.102 TCP        60 80 → 1161 [ACK] Seq=1 Ack=164091 Win=62780 Len=0
▶ Frame 192: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
▶ Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
▶ Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
▼ Transmission Control Protocol, Src Port: 1161, Dst Port: 80, Seq: 156469, Ack: 1, Len: 1460
    Source Port: 1161
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 1460]
    Sequence number: 156469    (relative sequence number)
    [Next sequence number: 157929    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  ▶ Flags: 0x010 (ACK)
    Window size value: 17520
    [Calculated window size: 17520]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0x34fc [unverified]
    [Checksum Status: Unverified]
```

TCP – RTT chart



Charts

- Each protocol has its own charts
- Useful to have a graphical representation of what's going on
- Typically exercises traces have "near-perfect" behavior
- Let's see what happens on a "real" trace

Downloading an Ubuntu ISO