Tutorial 1: Ping, Traceroute, Wireshark

TA: Katerina Lionta (klionta@csd.uoc.gr)  
cs-335a
Topics

- Ping
- Traceroute (Tracert)
- Wireshark
Internet: A global network
Ping

https://youtu.be/IlicPE38O-s?si=yQLviipmIXEqaFVt
Ping

- A tool for troubleshooting network issues, such as:
  - Test network connectivity (local network, Internet)
  - Test network Interface card
- Available in Windows, Linux, MacOS
- The sender sends 4 packets to the destination and waits for reply
- **RTT**(Round-Trip-Time): the time between the transmission of a packet from the transmitter until the reply from the receiver returns
Ping

- There is connectivity between two hosts, without packet losses
- 4 packets were sent, 4 replies were received
- You can ping an IP address or a server name
Ping

- No reply
- There isn’t connectivity between two hosts
- 4 packets were sent, 0 packets were received
- Possible reasons:
  - The receiver is power down
  - Firewall
Ping

- Not all the data packets reply back to the sender
- **Packet loss**
- Possible reasons:
  - Network congestion
  - Faulty hardware (cables, wiring, network card, modem)
Ping

- The route to the destination cannot be found
- A router doesn’t have any information on how to route data to the destination
- The destination is disconnecting from the Internet
Ping

C:\WINDOWS\system32>ping google.com

Pinging google.com [142.250.184.142] with 32 bytes of data:
Reply from 142.250.184.142: bytes=32 time=36ms TTL=110
Reply from 142.250.184.142: bytes=32 time=36ms TTL=110
Reply from 142.250.184.142: bytes=32 time=36ms TTL=110
Reply from 142.250.184.142: bytes=32 time=45ms TTL=110
Reply from 142.250.184.142: bytes=32 time=38ms TTL=110

Ping statistics for 142.250.184.142:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 36ms, Maximum = 45ms, Average = 38ms

Host's IP
RTT
Time To Live: Hops before the packet expires
Min RTT
Max RTT
Average RTT
Traceroute (tracert)

https://youtu.be/up3bcBLZS74?si=3zoCpaYqv6qA1A
Lu
Traceroute

- Used to show the route that the data packets take
- A tool that is used to find the exact path a data packet is taken from the sender to the receiver
- Available in Windows (tracert), UNIX and MacOS
- Pings every router in the path
- Sends three packets and measures the RTTs that the data packet took from each router and the destination
Traceroute

- **TTL**(Time-to-Live): how long the packet can live before it discarded, the maximum number of hops that packet can cross until it returns
- Every time a packet passes through a router, the router decreases the TTL by 1
- If TTL=0, the router drops the packet and a reply is transmitted to the sender that identifies the router
Traceroute

- The RTT of each of three packets from the sender to each router and the destination

<table>
<thead>
<tr>
<th>Number of hops</th>
<th>1st packet times</th>
<th>2nd packet times</th>
<th>3rd packet times</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;1 ms</td>
<td>&lt;1 ms</td>
<td>&lt;1 ms</td>
</tr>
<tr>
<td>2</td>
<td>8 ms</td>
<td>7 ms</td>
<td>8 ms</td>
</tr>
<tr>
<td>3</td>
<td>8 ms</td>
<td>8 ms</td>
<td>9 ms</td>
</tr>
<tr>
<td>4</td>
<td>9 ms</td>
<td>9 ms</td>
<td>fl.pompano.comcast.16.2.151.122.2</td>
</tr>
<tr>
<td>5</td>
<td>11 ms</td>
<td>12 ms</td>
<td>10 ms</td>
</tr>
<tr>
<td>6</td>
<td>12 ms</td>
<td>14 ms</td>
<td>miami.fl.libone.68.86.9.7</td>
</tr>
<tr>
<td>7</td>
<td>15 ms</td>
<td>17 ms</td>
<td>18 ms</td>
</tr>
<tr>
<td>8</td>
<td>20 ms</td>
<td>21 ms</td>
<td>22 ms</td>
</tr>
</tbody>
</table>

Trace complete.
Traceroute

- possible reasons:
  - problem with the specific router
  - not configured to return traceroute replies
- The packets passed to the next router
Traceroute

- Sets the TTL=4
- When the packet traverses 4 hops, it is dropped
Wireshark
Wireshark

- Packet sniffer: tool used for capturing network packets
- Download:
  - Windows and Macos:
    - [https://www.wireshark.org/download.html](https://www.wireshark.org/download.html)
    - In windows run the Wireshark as administrator
  - Linux:
    - [https://linuxhint.com/install_configure_wireshark_ubuntu/](https://linuxhint.com/install_configure_wireshark_ubuntu/)
Wireshark: Setup

- Run the Wireshark as administrator
- Select an active interface
Wireshark: Capturing

- Start capturing packets
- Stop capturing
Wireshark: Export files

- File > Export Packet
- Dissections > AS CSV...
### Wireshark

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>88</td>
<td>8.506952</td>
<td>192.168.2.8</td>
<td>185.125.188.55</td>
<td>TLSv.. 134 Change Cipher Spec, Application Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>8.506982</td>
<td>192.168.2.8</td>
<td>185.125.188.55</td>
<td>TLSv.. 1262 Application Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>8.507001</td>
<td>192.168.2.8</td>
<td>185.125.188.55</td>
<td>TCP 1514 51134 -&gt; 443 [PSH, ACK] Seq=1553 Ack=3554 Win=130560</td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>8.507015</td>
<td>192.168.2.8</td>
<td>185.125.188.55</td>
<td>TLSv.. 1514 Application Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>8.508394</td>
<td>192.168.2.8</td>
<td>185.125.188.55</td>
<td>TLSv.. 1108 Application Data, Application Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>93</td>
<td>8.575764</td>
<td>185.125.188.55</td>
<td>192.168.2.8</td>
<td>TLSv.. 133 Application Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>8.575764</td>
<td>185.125.188.55</td>
<td>192.168.2.8</td>
<td>TLSv.. 133 Application Data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>8.575852</td>
<td>192.168.2.8</td>
<td>185.125.188.55</td>
<td>TCP 54 51134 -&gt; 443 [ACK] Seq=5527 Ack=3712 Win=130560 Len=0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>8.585799</td>
<td>185.125.188.55</td>
<td>192.168.2.8</td>
<td>TCP 60 443 -&gt; 51134 [ACK] Seq=3712 Ack=2997 Win=59520 Len=0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>8.592751</td>
<td>185.125.188.55</td>
<td>192.168.2.8</td>
<td>TCP 60 443 -&gt; 51134 [ACK] Seq=3712 Ack=4473 Win=58112 Len=0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>8.640895</td>
<td>185.125.188.55</td>
<td>192.168.2.8</td>
<td>TCP 60 443 -&gt; 51134 [ACK] Seq=3712 Ack=5527 Win=57088 Len=0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>8.757826</td>
<td>185.125.188.55</td>
<td>192.168.2.8</td>
<td>TCP 1498 443 -&gt; 51134 [ACK] Seq=3712 Ack=5527 Win=57088 Len=1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>8.760040</td>
<td>185.125.188.55</td>
<td>192.168.2.8</td>
<td>TCP 1498 443 -&gt; 51134 [PSH, ACK] Seq=5527 Ack=5527 Win=57088</td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>8.760040</td>
<td>185.125.188.55</td>
<td>192.168.2.8</td>
<td>TCP 1498 443 -&gt; 51134 [ACK] Seq=6600 Ack=5527 Win=57088 Len=1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>8.760077</td>
<td>192.168.2.8</td>
<td>185.125.188.55</td>
<td>TCP 54 51134 -&gt; 443 [ACK] Seq=5527 Ack=8044 Win=131328 Len=0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>8.763258</td>
<td>185.125.188.55</td>
<td>192.168.2.8</td>
<td>TCP 1498 443 -&gt; 51134 [PSH, ACK] Seq=8044 Ack=5527 Win=57088</td>
<td></td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>8.763285</td>
<td>192.168.2.8</td>
<td>185.125.188.55</td>
<td>TCP 54 51134 -&gt; 443 [ACK] Seq=5527 Ack=9488 Win=131328 Len=0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>8.764356</td>
<td>185.125.188.55</td>
<td>192.168.2.8</td>
<td>TCP 1498 443 -&gt; 51134 [ACK] Seq=9488 Ack=5527 Win=57088 Len=1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **No.**: the serial number of the packet
- **Time**: the time of the transmission/receiving of the packet (starts from 0, the moment that the capturing started) in seconds
- **Source**: the source IP address
- **Destination**: the destination IP address
- **Protocol**: the protocol used
- **Length**: the length of the packets in bytes
- **Info**: extra information about the packet (header fields, flags etc)
Wireshark: Filtering

- You can filter with:
  - Transfer protocol name (tcp, udp etc)
  - Source IP, destination IP
    (ip.src==192.168.0.0, ip.dst==192.168.0.0)
- You can use logical operators:
  - and, or
  - &&, ||, !
- Examples:
  - ip.src != 10.43.54.65 or ip.dst != 10.43.54.65
  - tcp
  - udp

https://wiki.wireshark.org/DisplayFilters
Wireshark: Encapsulation

- Encapsulation allows us to use different protocols in all levels of the TCP/IP stack.
- Wireshark shows us the headers of all these levels (e.g. an HTTP packet)
Wireshark: Encapsulation

- Five headers (Physical layer, Link layer, Network layer, Transport layer, App layer)

Expand header
Wireshark: Packet details

As mentioned, you can see the details of each package by clicking on it. You can double-click to open it in a new window. There you can see the packet's details, as they're shown on the previous slide. You can also see the hex format of the packet:
Wireshark: Packet details

You can hover over the bytes and see what they represent (see bottom left corner):
MY FACE AFTER

RUNNING WIRESHARK FOR THE FIRST TIME