Computer Networks and Communication

Lecture 2

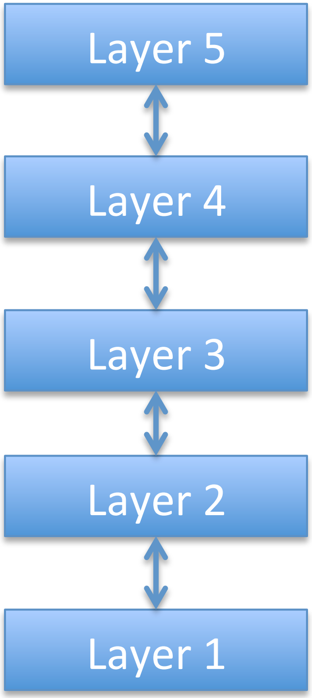
Layered Network Architecture

Network-Design Issues

- I design an application, say, a web browser, do I have to care what kind of networks my software will run on?
- In other words, do I have to care if the user will use my browser on Ethernet network (IEEE 802.3) or WiFi (IEEE 802.11n)?
- And if I design web protocol, say **Hypertext Transfer Protocol (HTTP)**, do I have to concern these issue?
  - Would my protocol be compatible with aforementioned protocols or any other protocols?
  - Would it work on all kinds of hardware?

Network Layers

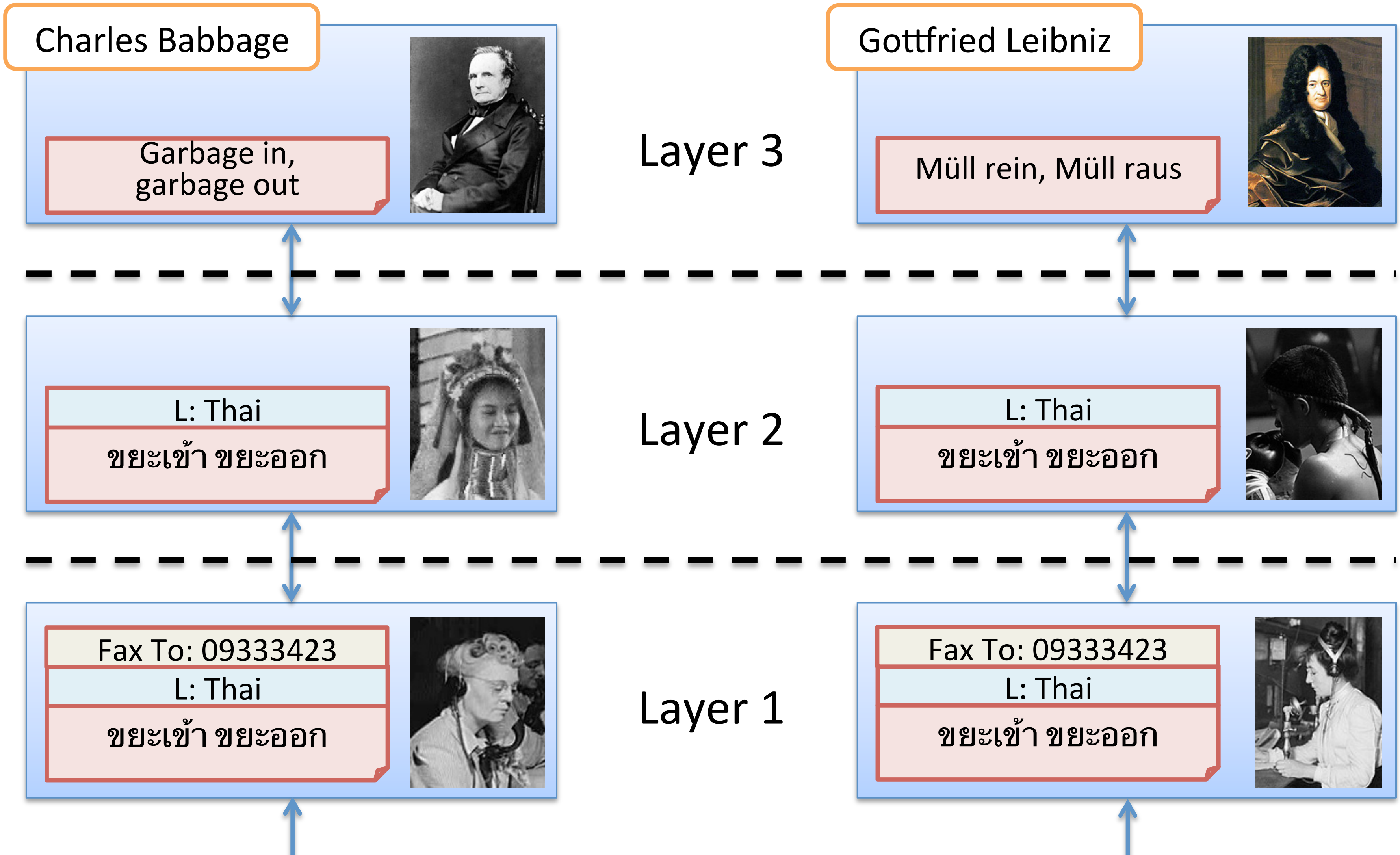
- To reduce design complexity, most networks are organized as ***stacks*** of **layers** or **levels**
- Each layer:
  - Is built upon the one below it
  - Contents and function of each layer differ from network to network
  - Provides certain **services** to the higher layers
  - Shielding the other layers from the details of how the services are implemented



Layers Analogy

- Two mathematicians want to talk to each other
  - One speaks English
  - One speaks German
- Because they do not have common language, each of them has a translator
- Both translators understand Thai, so they agree to use Thai as their common language
  - English <-> Thai
  - German <-> Thai
- Also, they both have secretaries who are responsible to send messages to each other
  - They normally send faxes to each other

Layers Analogy (2)



Layers Analogy (3)

- What if Charles Babbage wants to talk to another mathematician, say, Joseph Fourier:
  - Does he has to change the translator?

    No, as long as Fourier's translator speaks Thai
    That is, French <-> Thai
  - If he changes the translator, does he has to change the secretary too?

    No
  - What if the translators change their common language?
- If the secretary wants to use email instead of fax, would it affect the translators or the mathematicians

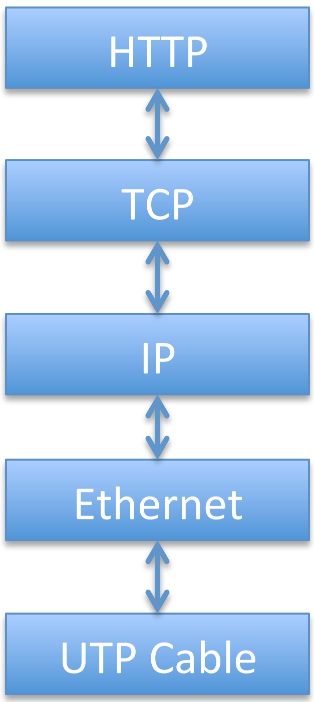
    No, however secretaries at both sides (sender and receiver) must agree to use email

Layers Analogy (4)

- As we have seen from the example, each layer is separated from each other
- The lower layers offer services for upper ones
- One can alter or change any layer without any affect to other layers
- Nevertheless, if we want to change something within the same layer, the other parties has to acknowledge and agree on the change

Protocol Stack

- As mentioned in Slide 4, a computer network architecture composed of layers
- Each layer offers services to the higher layers
- A service in each layer must use the same protocol
- A set of protocols that work together in a network, **one protocol per layer**, is called a **protocol stack**

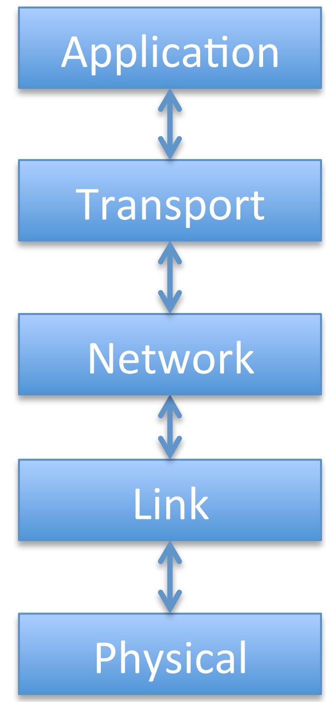


An example of a protocol stack

TCP/IP Protocol Stack

- The TCP/IP or the Internet protocol stack consists of 5 layers
  1. Physical layer
  2. Link layer
  3. Network layer
  4. Transport layer
  5. Application layer

REMEMBER ALL OF THEM!!



- This reference protocol stack is sometimes referred to as **TCP/IP Reference Model**

- The layers are sorted from the lowest to the highest one

Application Layer

- This is the layer where network applications and their application layer protocol reside
- Example: HTTP, SMTP, FTP, DNS, Skype
- An application-layer protocols is distributed among end systems
- An application at an end systems use that protocol to exchange pieces of information with another application at another end system
- We refer to this piece of information as a **message**

Transport Layer

- Protocols in this layer are responsible for transporting messages between applications at two end systems
- There are two transport protocols in TCP/IP architecture:
  - **Transmission Control Protocol (TCP)**
  - **User Datagram Protocol (UDP)**
- We refer to a transport-layer packet as a **segment**

Transport Layer - TCP

- TCP: Transmission Control Protocol
- It provides
  - **Guaranteed delivery:** application-layer messages are guaranteed to arrive correctly, complete and in-order
  - **Flow control:** prevent sender from sending the data faster than the receiver can handle
  - **Congestion control**: The source adapts its transmission rate when the network is congested

Transport Layer - UDP

- UDP: User Datagram Protocol
- It does not provide anything
  - **Unreliable transmission**: The received message could contain errors or out of order. Some messages could be lost along the way
  - **No flow and congestion controls**: Sender just keep sending the messages thinking about the receiver or network traffic
- Question:
  - Why would we need such transmission service?
  - What are benefits of UDP over TCP?
  - Are there any applications that want to send data which might be erroneous? Why would they want to do so?

Network Layer

- In the Internet, the primary protocol in this layer is the **Internet Protocol (IP)**
- It is responsible for
  - **Host addressing**: Assigning an address to a network device. This is a logical address which is recognized by all devices in the network
  - **Routing**: Deliver packets **from the source to the destination** based on the address
- Currently, there are two versions of IP: **IPv4** and **IPv6**
- Main difference between the two versions is the **address space**
  - IPv4 address space: **32 bits** / 2^{32} = ~4.29 Billion addresses
  - IPv6 address space: **128 bits** / 2^{128} = ~ 3.4×10^{38} addresses

Network Layer (2)

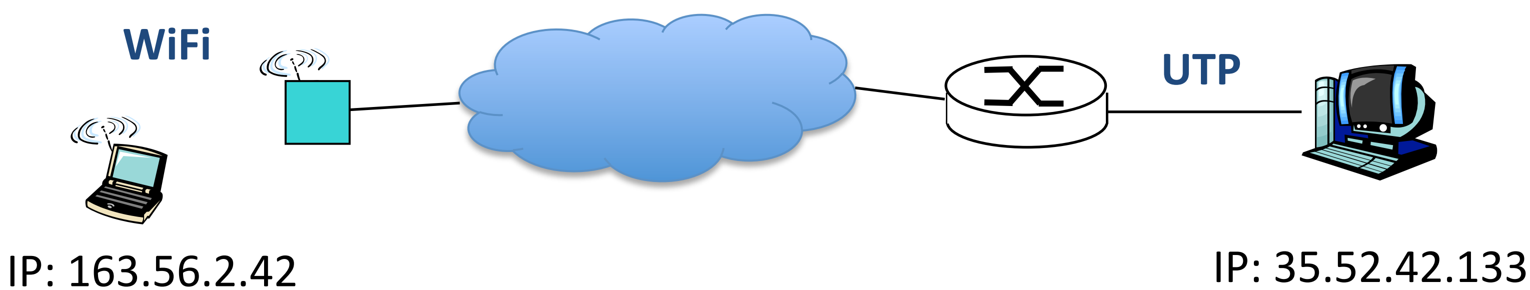
- Packet delivery is carried out based only on the address specified on the packet
  - IP does not need any traffic control or monitoring facility
  - Each router along the path simply forwards a packet to the next router using a **routing protocol**
- Packets at this layer are referred to as **datagrams**
- The Internet Protocol handles each datagram independent from each other
  - It does not concern the order of the packets which belong to the same connection.
  - That task is carried out by transport-layer protocol

Link Layer

- Provides reliable data delivery to neighboring device through **one** physical link
  - **Media access control**: Host addressing and media access mechanism
  - **Logical link control**: Multiplexing mechanism
  - **Error detection**
- For example, link layer provides data transmission
  - from N' Pattie's laptop to her WiFi router
  - from a router to the next router

Link Layer (2)

- Protocol in this layer depends directly on the physical medium of each link, e.g.,
  - IEEE 802.11n for WiFi radio link
  - IEEE 802.3 for unshielded twisted pair (UTP) cables
- That is, link-layer protocols operate in only one network, and not *inter*-network like TCP or IP
- Link-layer packets are called **frames**



Physical Layer

- This layer is where actual data signals are sent through a transmission medium

- They specify how to send **individual bits** within a link-layer frame from one node to the next
  - How many volts should represent 1 and 0?
  - How many nanosecond a bit should last?

- Indeed, these protocols are dependent on the type of the medium

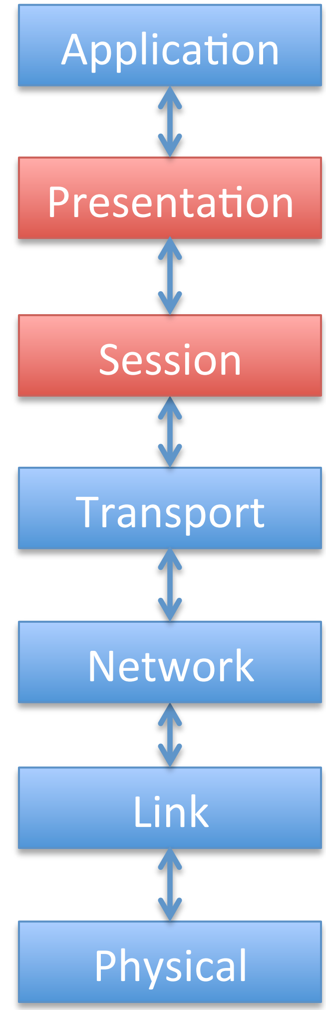
OSI Reference Model

- A **network reference model** is a model that describes a network architecture
- We have just discussed **TCP/IP reference model**, which consists of 5 layers
- There exist another model, which is widely used: **Open Systems Interconnection (OSI) Model**
- It is defined by International **Organization for Standardization (ISO)** in 1970s
- Sometimes it is referred to as **ISO OSI model**

OSI Reference Model (2)

- OSI Model consists of 7 layers
  1. Physical layer
  2. Link layer
  3. Network layer
  4. Transport layer
  5. **Session layer**
  6. **Presentation layer**
  7. Application layer

REMEMBER THEM TOO!!



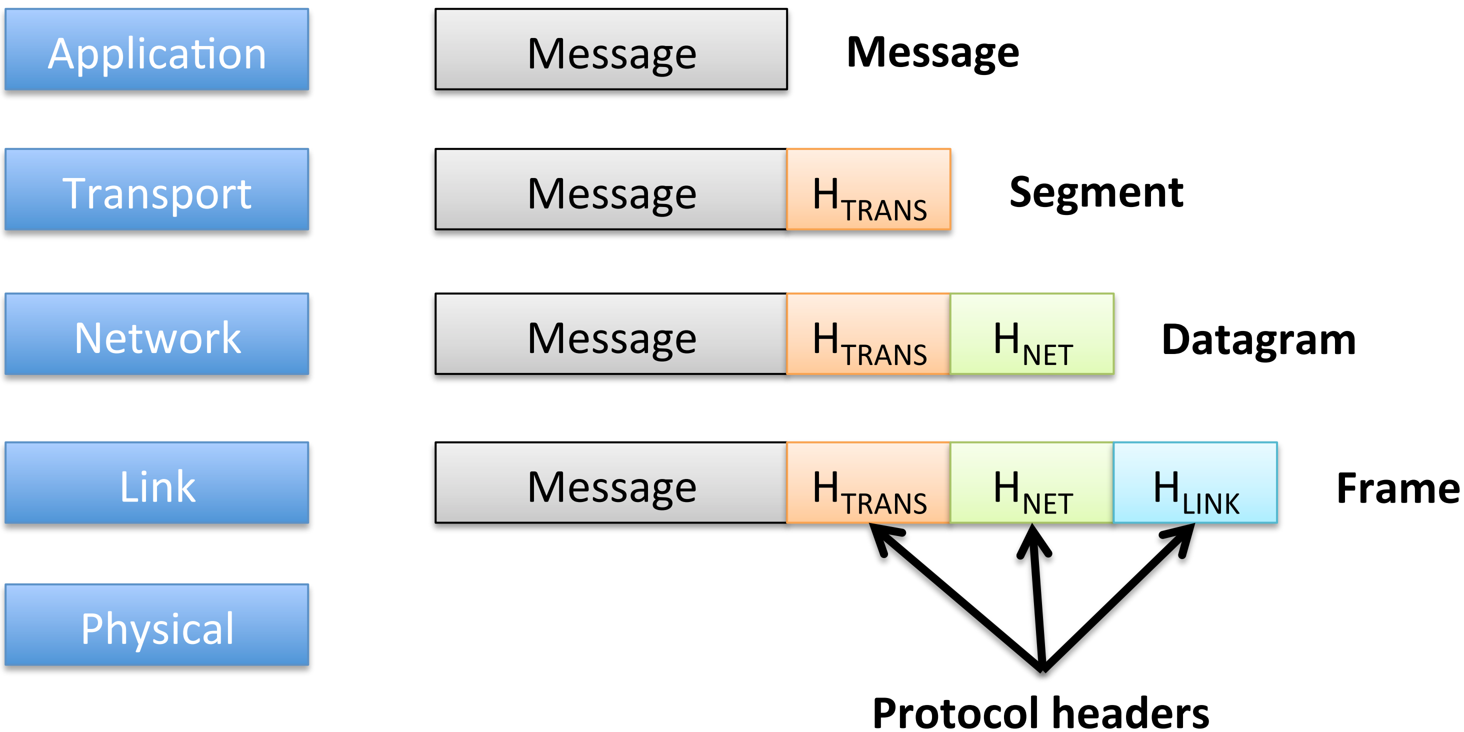
OSI Reference Model

- Session layer
  - Allows users on different machines to establish sessions between them (e.g., keeping track of transmission order)
  - Provides synchronization of data exchange (e.g., if the communication is lost, the protocol knows where to continue the transmission)
- Presentation layer
  - Is concerned with data interpretation of the transmitted information
  - Might also provide data encryption and compression
- Both layers are not included in the TCP/IP model
  - If an application needs any of their functionalities, the application developers should implement them themselves

Encapsulation

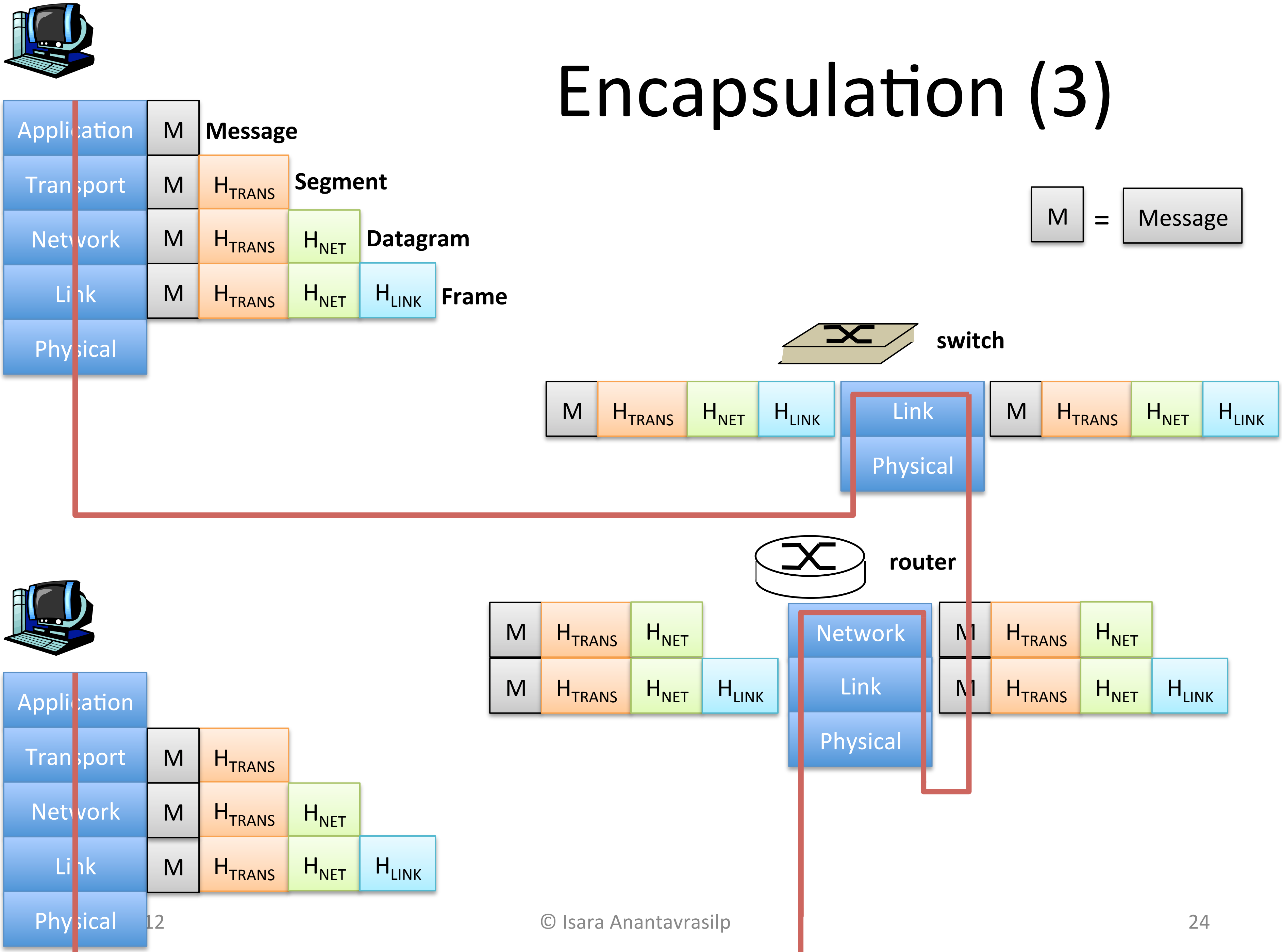
- When a packet is sent from a source to a destination
  - IP routers along the path concern only the how to route the packet based on the IP address
  - They do not implement all of the layers in the protocol stack
- Similarly, link-layer devices (such as Ethernet switches)
  - Do not even recognize the IP address.
  - They recognize only Ethernet addresses
- As you can see, devices at each layer concern only the information that are relevant to them. Other information in the packet is ignored
- This leads to a **very important** concept of **encapsulation**

Encapsulation (2)



- At each layer, a packet has two types of fields:
  - **Header field**: Information of the packet used by the protocol at that layer
  - **Payload field**: The packet conveyed from the layer above

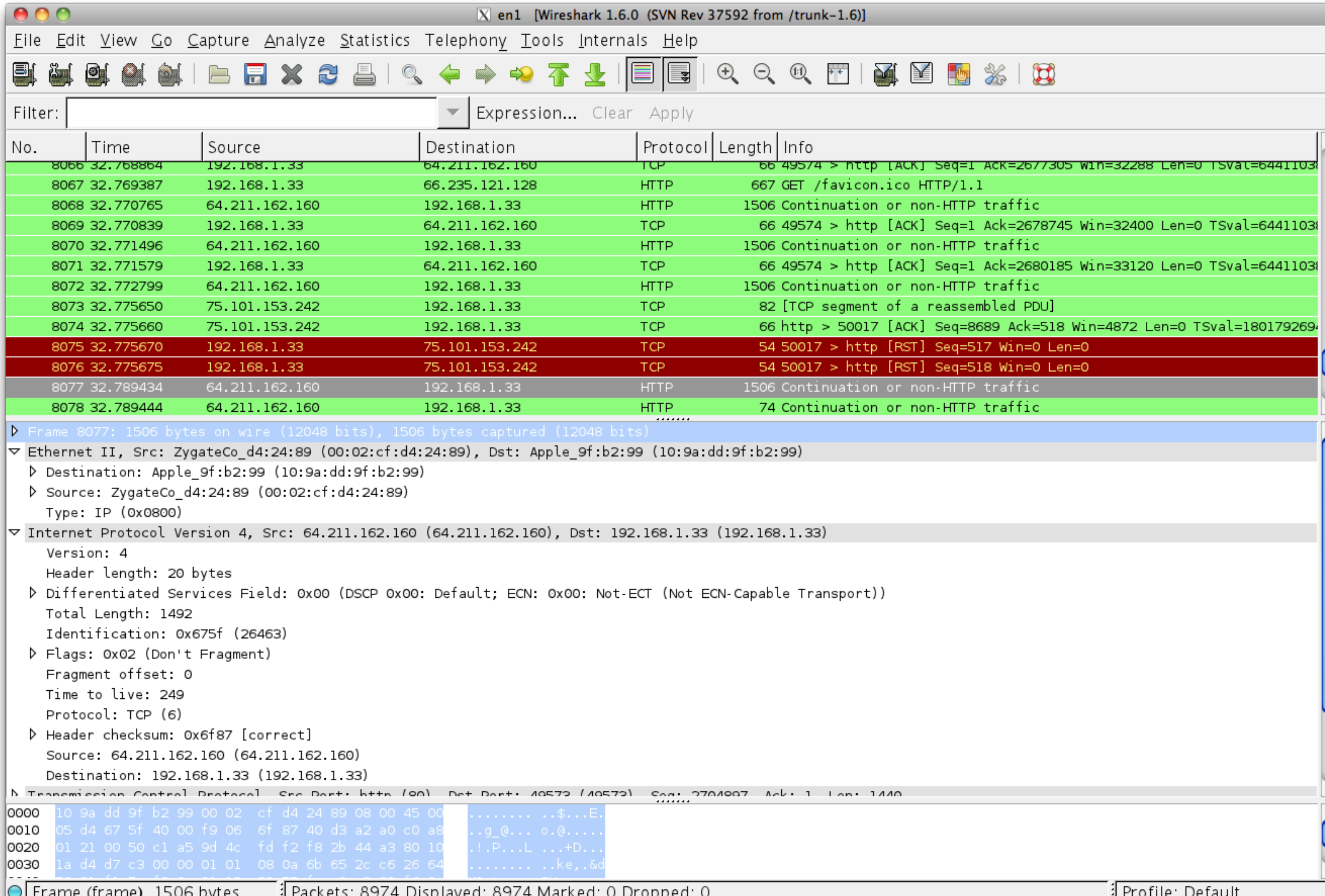
Encapsulation (3)



Wireshark

- Now, let us have a look into actual IP packets and see how they look like
- We will use a software called **Wireshark**
  - It is a packet **sniffer** and analyzer software
  - It can passively capture and record packets passing over a network
  - It can look into the content of each packet
  - Open-source
  - Available in many OSes, including MS Windows and MacOS
  - See: http://www.wireshark.org/

Wireshark (2)



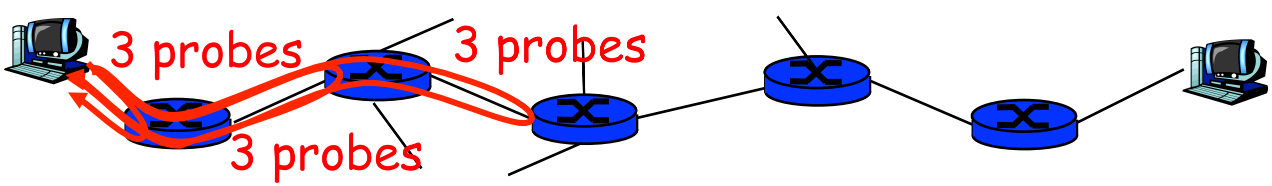
en1 [Wireshark 1.6.0 (SVN Rev 37592 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply
No. Time Source Destination Protocol Length Info
8067 32.769387 192.168.1.33 66.235.121.128 HTTP 667 GET /favicon.ico HTTP/1.1
8068 32.770765 64.211.162.160 192.168.1.33 HTTP 1506 Continuation or non-HTTP traffic
8069 32.770839 192.168.1.33 64.211.162.160 TCP 66 49574 > http [ACK] Seq=1 Ack=2678745 Win=32400 Len=0 TSval=6441103
8070 32.771496 64.211.162.160 192.168.1.33 HTTP 1506 Continuation or non-HTTP traffic
8071 32.771579 192.168.1.33 64.211.162.160 TCP 66 49574 > http [ACK] Seq=1 Ack=2680185 Win=33120 Len=0 TSval=6441103
8072 32.772799 64.211.162.160 192.168.1.33 HTTP 1506 Continuation or non-HTTP traffic
8073 32.775650 75.101.153.242 192.168.1.33 TCP 82 [TCP segment of a reassembled PDU]
8074 32.775660 75.101.153.242 192.168.1.33 TCP 66 http > 50017 [ACK] Seq=8689 Ack=518 Win=4872 Len=0 TSval=18017926
8075 32.775670 192.168.1.33 75.101.153.242 TCP 54 50017 > http [RST] Seq=517 Win=0 Len=0
8076 32.775675 192.168.1.33 75.101.153.242 TCP 54 50017 > http [RST] Seq=518 Win=0 Len=0
8077 32.789434 64.211.162.160 192.168.1.33 HTTP 1506 Continuation or non-HTTP traffic
8078 32.789444 64.211.162.160 192.168.1.33 HTTP 74 Continuation or non-HTTP traffic
Frame 8077: 1506 bytes on wire (12048 bits), 1506 bytes captured (12048 bits)
Ethernet II, Src: ZygateCo_d4:24:89 (00:02:cf:d4:24:89), Dst: Apple_9f:b2:99 (10:9a:dd:9f:b2:99)
Destination: Apple_9f:b2:99 (10:9a:dd:9f:b2:99)
Source: ZygateCo_d4:24:89 (00:02:cf:d4:24:89)
Type: IP (0x0800)
Internet Protocol Version 4, Src: 64.211.162.160 (64.211.162.160), Dst: 192.168.1.33 (192.168.1.33)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
Total Length: 1492
Identification: 0x675f (26463)
Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to live: 249
Protocol: TCP (6)
Header checksum: 0x6f87 [correct]
Source: 64.211.162.160 (64.211.162.160)
Destination: 192.168.1.33 (192.168.1.33)
0000 10 9a dd 9f b2 99 00 02 cf d4 24 89 08 00 45 00 ....$...E.
0010 05 d4 67 5f 40 00 f9 06 6f 87 40 d3 a2 a0 c0 a8 ..g_@... o.@.....
0020 01 21 00 50 c1 a5 9d 4c fd f2 f8 2b 44 a3 80 10 .!.P...L ...+D...
0030 1a d4 d7 c3 00 00 01 01 08 0a 6b 65 2c c6 26 64 ....ke,.&d
Frame (frame), 1506 bytes
Packets: 8974 Displayed: 8974 Marked: 0 Dropped: 0
Profile: Default

Traceroute

- Let us see how real Internet delay and packet loss look like!!
- **Traceroute**
  - Determines and displays route of the packets from a host to another host
  - It also measures the **round-trip** delays to all intermediate routers along the way
  - Try it yourself at: http://www.traceroute.org/

Traceroute (2)

- How Traceroute works:
  - It sends three packets to each router along the path
  - For each probing packet, routers reply back with another packet
  - Traceroute determines which routers send the replies back and how do the replies take



Traceroute (3)

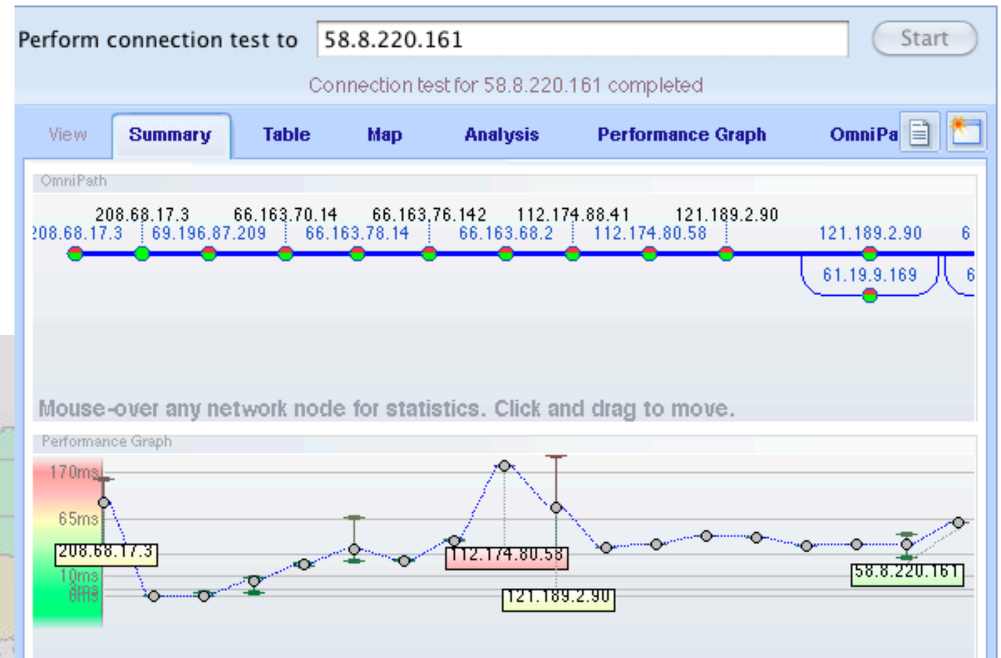
```
Traceroute from EUNet, Latvia, to ppp-58-8-220-161.revip2.asianet.co.th
1 194.8.7.194 (194.8.7.194) 0.256 ms 0.147 ms 0.136 ms
2 tl-bgp22-fe-0-0-0-66.telia.lv (194.19.226.233) 0.339 ms 0.335 ms 0.321 ms
3 hls-b2-link.telia.net (80.239.193.249) 8.474 ms 8.430 ms 8.458 ms
4 s-bb1-link.telia.net (80.91.246.84) 15.477 ms 15.455 ms s-bb1-link.telia.net (80.91.251.33) 15.406 ms
5 ffm-bb1-link.telia.net (80.91.248.53) 42.338 ms ffm-bb1-link.telia.net (80.91.246.210) 42.356 ms 42.391
ms
6 prs-bb1-link.telia.net (80.91.245.105) 52.263 ms prs-bb1-link.telia.net (80.91.245.101) 52.304 ms prs-
bb1-link.telia.net (80.91.245.103) 119.066 ms
7 snge-b2-link.telia.net (80.91.245.153) 248.099 ms 248.115 ms 248.072 ms
8 hnk-b2-link.telia.net (80.91.245.149) 280.203 ms hnk-b2-link.telia.net (80.91.245.151) 278.606 ms hnk-
b2-link.telia.net (80.91.245.149) 280.198 ms
9 cat-ic-143725-hnk-b2.c.telia.net (80.239.167.2) 278.582 ms cat-ic-143727-hnk-b2.c.telia.net
(213.248.86.122) 278.537 ms cat-ic-143725-hnk-b2.c.telia.net (80.239.167.2) 281.855 ms
10 61.19.9.169 (61.19.9.169) 330.657 ms 333.791 ms 333.891 ms
11 61.19.9.153 (61.19.9.153) 332.285 ms 332.334 ms 332.347 ms
12 61.19.9.34 (61.19.9.34) 332.353 ms 332.348 ms 330.614 ms
13 61.19.15.106 (61.19.15.106) 333.011 ms 333.841 ms 349.869 ms
14 61-91-210-6.static.asianet.co.th (61.91.210.6) 334.668 ms 331.379 ms 331.361 ms
15 203-144-144-8.static.asianet.co.th (203.144.144.8) 335.086 ms 333.479 ms 333.442 ms
16 61-91-210-49.static.asianet.co.th (61.91.210.49) 335.027 ms 61-91-210-52.static.asianet.co.th
(61.91.210.52) 333.406 ms 61-91-210-49.static.asianet.co.th (61.91.210.49) 334.962 ms
17 119-46-176-134.static.asianet.co.th (119.46.176.134) 364.269 ms 339.948 ms
119-46-78-142.static.asianet.co.th (119.46.78.142) 341.656 ms
18 * * *
19 ppp-58-8-220-161.revip2.asianet.co.th (58.8.220.161) 367.860 ms 354.296 ms 355.440 ms
```

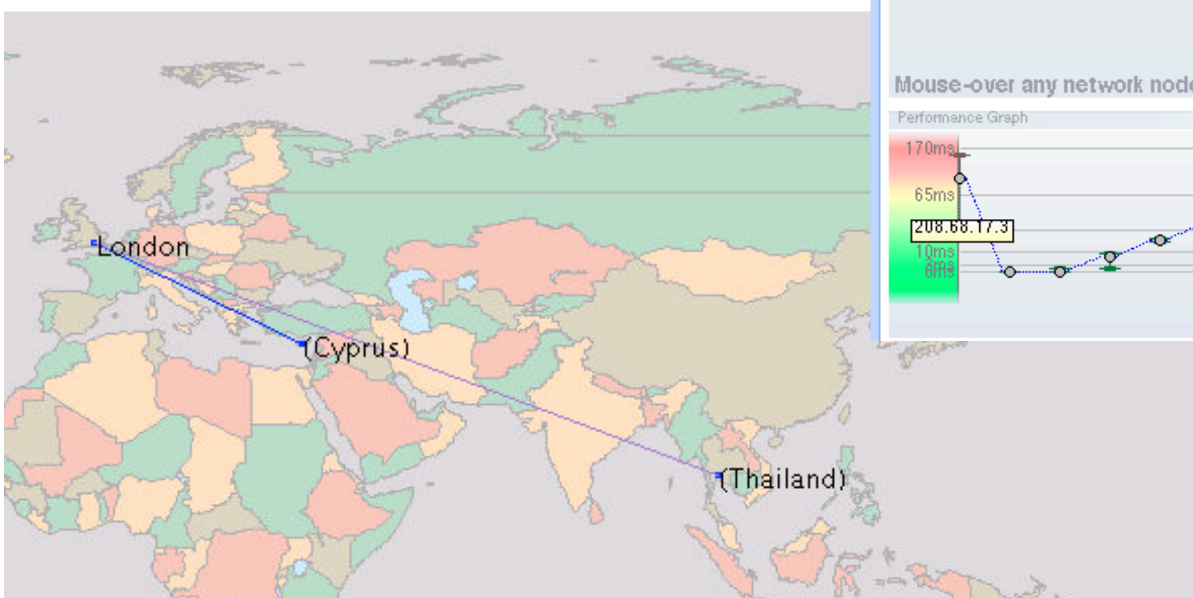
Traceroute (4)

```
1 194.8.7.194 (194.8.7.194) 0.256 ms 0.147 ms 0.136 ms

2 tl-bgp22-fe-0-0-0-66.telia.lv (194.19.226.233) 0.339 ms
0.335 ms 0.321 ms

3 hls-b2-link.telia.net (80.239.193.249) 8.474 ms 8.430 ms
8.458 ms

4 s-bb1-link.telia.net (80.91.246.84) 15.477 ms 15.455 ms
 s-bb1-link.telia.net (80.91.251.33) 15.406 ms

5 ffm-bb1-link.telia.net (80.91.248.53) 42.338 ms
ffm-bb1-link.telia.net (80.91.246.210) 42.356 ms 42.391 ms
```

VisualRoute

- Visualized route tracer with additional network analyzing tools
- http://www.visualware.com/





Further Reading

- Briley, K., *IPv4 vs IPv6 – What Are They, Exactly?*, http://www.thetechlabs.com/tech-news/ipv4-vs-ipv6/
- *The TCP Guide*, http://www.tcpipguide.com/