CS4238:
Computer Security Practice

Lecture 2: Networking Overview & Configuration, Attack Framework, Reconnaissance

Slides by: LIANG Zhenkai,
Roland YAP & SUFATRIO

Outline

- Networking Overview
- Network Configuration: Linux desktop

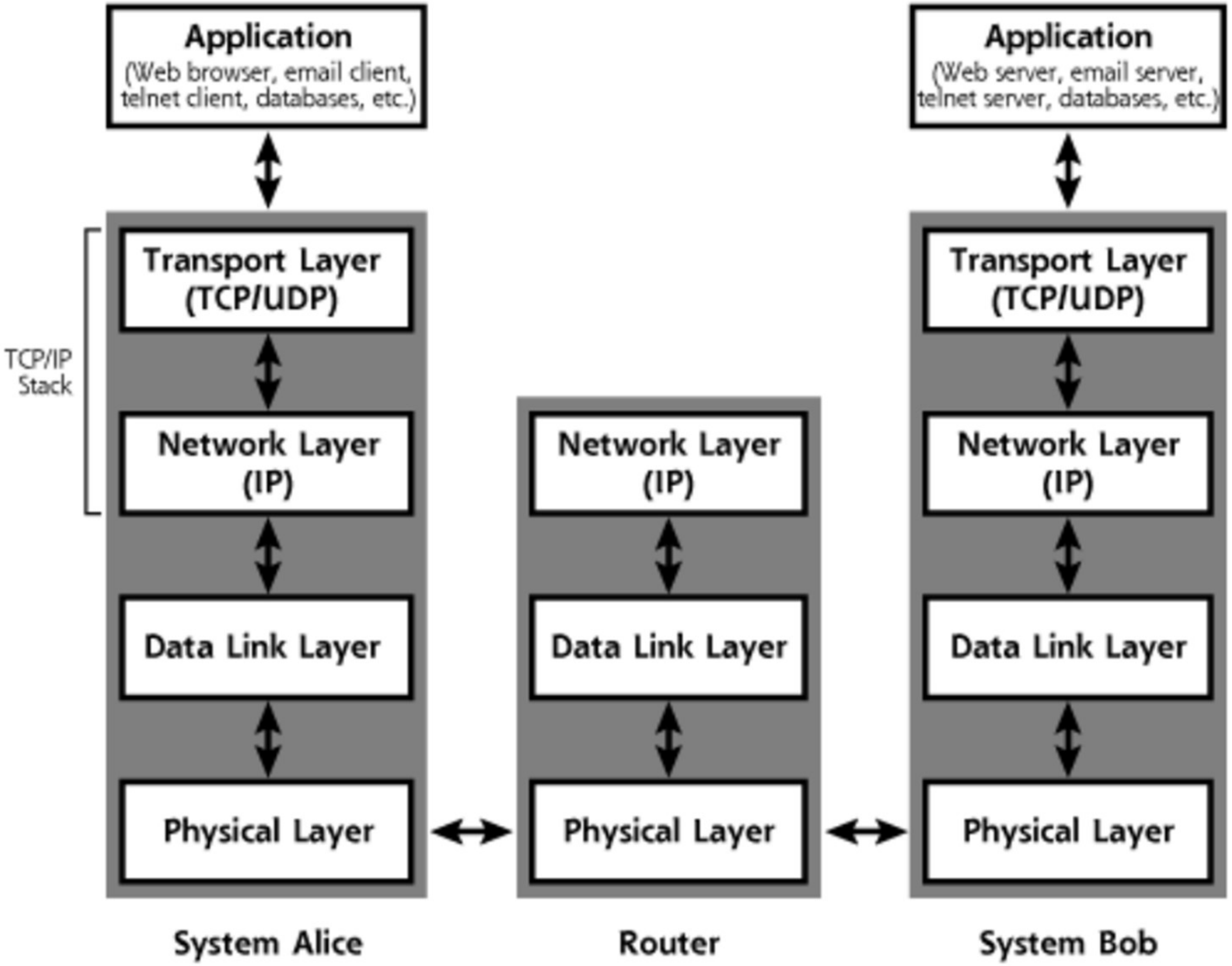
Networking Overview

(Chapter 2 of the Reference book 1)

Relevant Networking Concepts

- TCP/IP Layers
  - Application
  - Transport
  - Network
  - Data Link
  - Physical
- TCP and UDP
- IP and ICMP
- Routing
  - NAT
- Firewall
- Ethernet and 802.11
  - ARP
- SSL and TLS
- IPSec and VPN

TCP/IP Layers

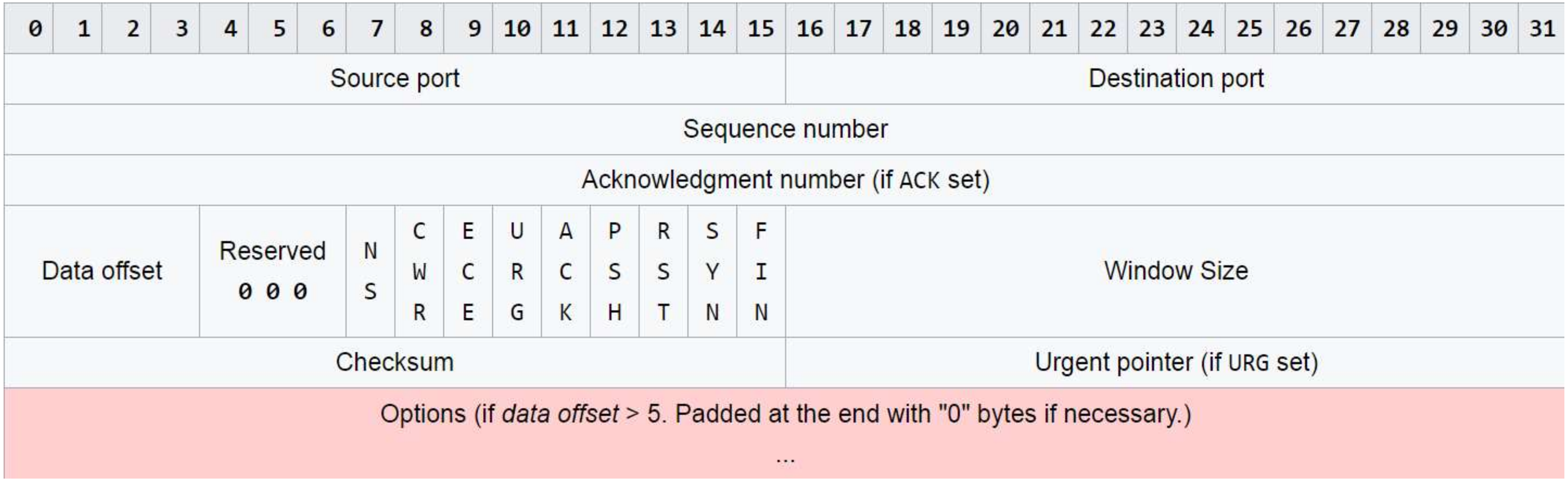


Source: Skoudis & Liston, Counter Hack Reloaded

Transport Layer

TCP

- TCP header format:

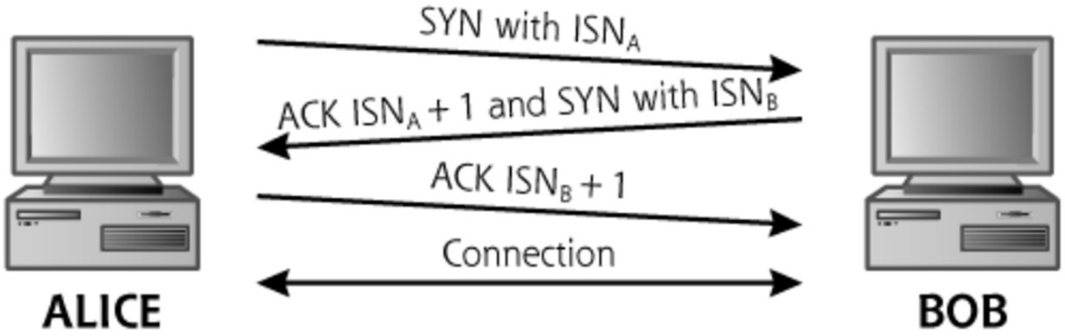


<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td colspan="16">Source port</td><td colspan="16">Destination port</td></tr>
<tr><td colspan="32">Sequence number</td></tr>
<tr><td colspan="32">Acknowledgment number (if ACK set)</td></tr>
<tr><td colspan="4">Data offset</td><td colspan="3">Reserved
0 0 0</td><td>N
S</td><td>C
W
R</td><td>E
C
E</td><td>U
R
G</td><td>A
C
K</td><td>P
S
H</td><td>R
S
T</td><td>S
Y
N</td><td>F
I
N</td><td colspan="16">Window Size</td></tr>
<tr><td colspan="16">Checksum</td><td colspan="16">Urgent pointer (if URG set)</td></tr>
<tr><td colspan="32">Options (if <i>data offset</i> > 5. Padded at the end with "0" bytes if necessary.)
...</td></tr>
</table>

Source: Wikipedia

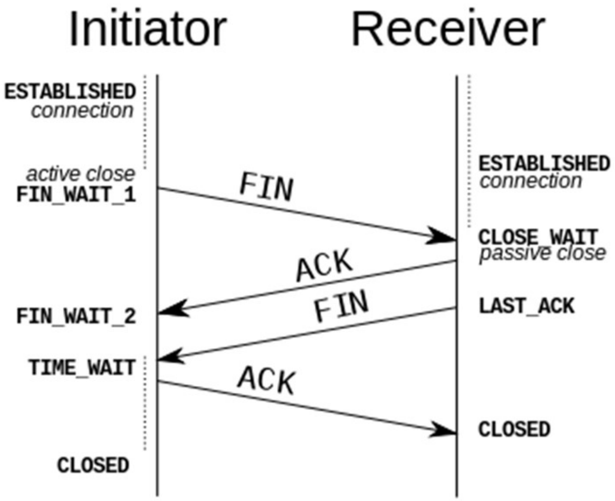
TCP Connection Management

- TCP three-way handshake:



Source: Skoudis & Liston,
Counter Hack Reloaded

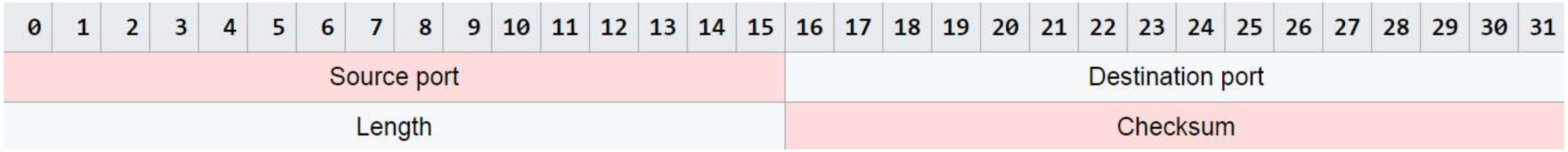
- TCP connection termination:



Source: Wikipedia

UDP

- Connectionless transport protocol
- UDP header format:



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Source port																Destination port															
Length																Checksum															

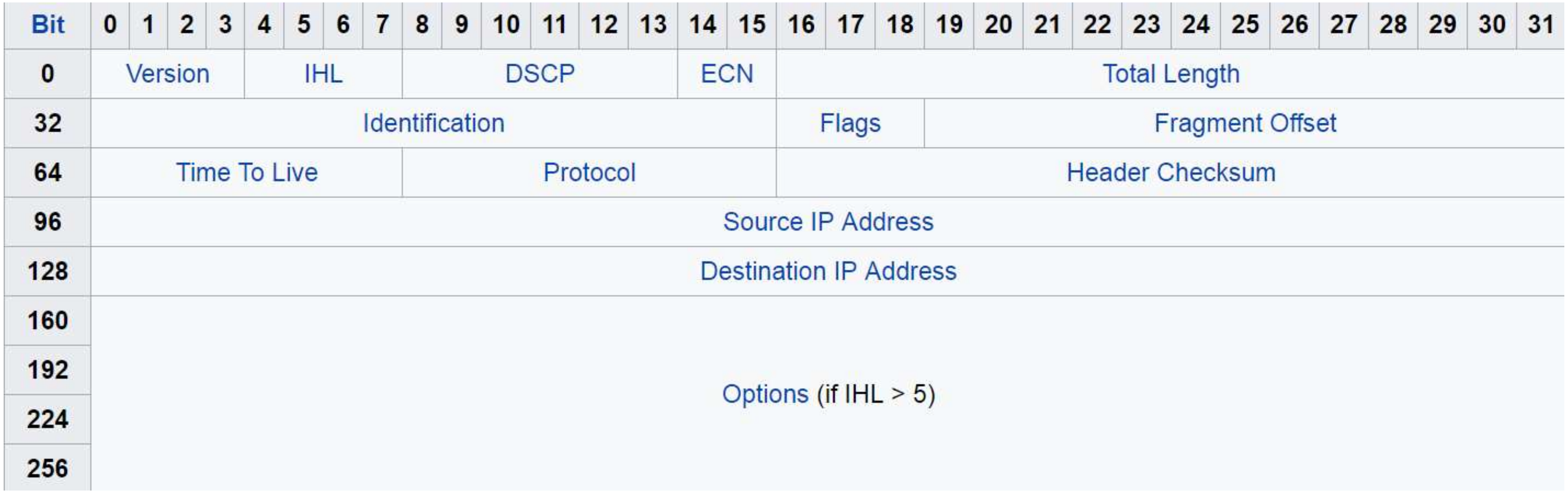
Source: Wikipedia

- Used among others by DNS (port 53), BOOTP/DHCP (port 67 & 68), TFTP (port 69), SNMP (port 161)

Network Layer

IP

- Importance of IP:
  - "Anything over IP and IP over anything"
  - The waist (glue point) of protocol-stack's hourglass
- IP header format:



Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Version				IHL				DSCP						ECN		Total Length															
32	Identification																Flags			Fragment Offset												
64	Time To Live								Protocol								Header Checksum															
96	Source IP Address																															
128	Destination IP Address																															
160	Options (if IHL > 5)																															
192																																
224																																
256																																

Source: Wikipedia

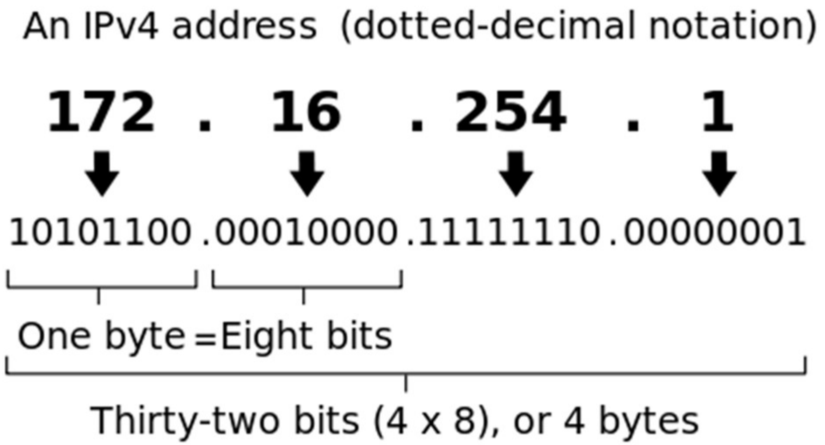
IP Packet Fragmentation & Reassembly

- **Goal**: To optimize packet length for various communication links with different maximum transmission unit (MTU)
- Two **flag bits** in IP header:
  - Don’t Fragment bit
  - More Fragment bit
- Other related IP **header fields**:
  - Identification: set to a unique value
  - Fragment Offset: where a fragment needs to be positioned during the reassembly

Source: Wikipedia

IPv4 Address

- Dotted-decimal notation:


An IPv4 address (dotted-decimal notation)
172 . 16 . 254 . 1
10101100.00010000.11111110.00000001
One byte = Eight bits
Thirty-two bits (4 x 8), or 4 bytes

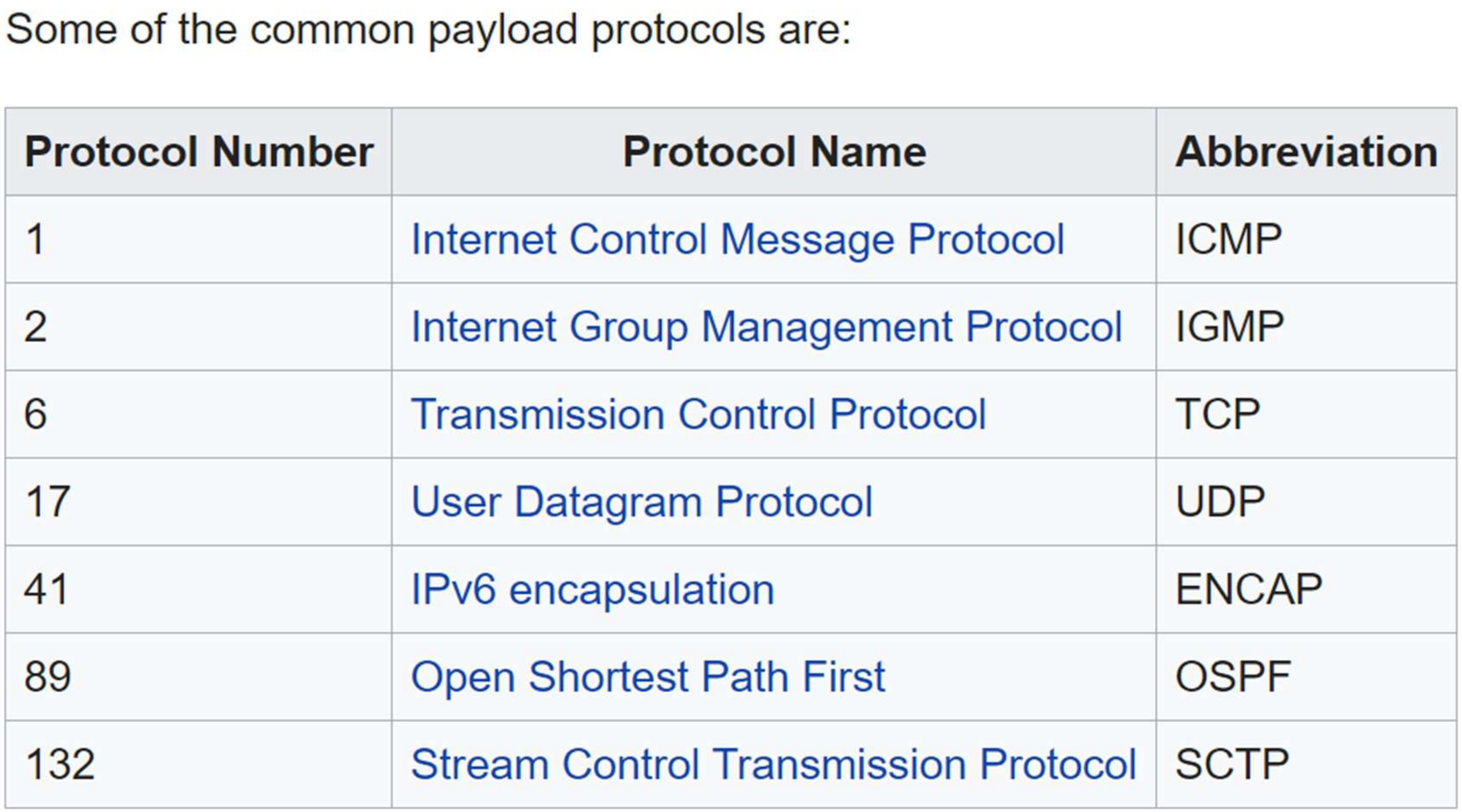
Source: Wikipedia

- Network address and host address components
- Classful network architecture (1981-1993):
  - Now only for default configuration of subnet masks
- Classless Inter-Domain Routing (CIDR):
  - Variable-length subnet masking (VLSM)
  - CIDR notation (e.g. 192.168.2.0/24)

IPv4 Address

- **Special IP** addresses:
  - Localhost address: 127.0.0.1
  - Private addresses:
    - **10.0.0.0 – 10.255.255.255**: 24-bit host ID (24-bit block)
    - **172.16.0.0 – 172.31.255.255**: 20-bit host ID (20-bit block)
    - **192.168.0.0 – 192.168.255.255**: 16-bit host ID (16-bit block)
    - Not routable on the public Internet
    - Usually used together with NAT or proxy
  - Automatic Private IP Addressing (APIPA) or auto-IP address: 169.254.1.0 – 169.254.254.255
    - E.g. when DHCP server is unavailable

Protocols on Top of IP



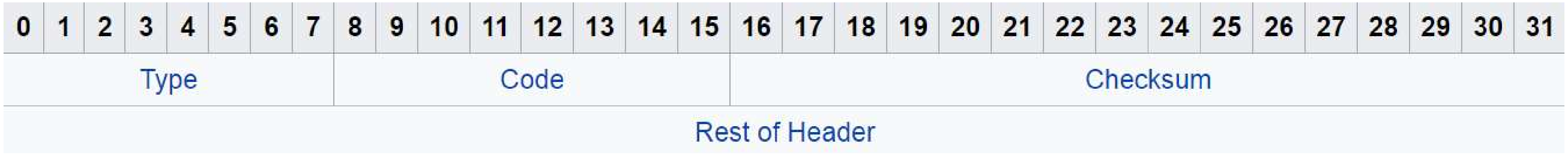
Some of the common payload protocols are:

Protocol Number	Protocol Name	Abbreviation
1	Internet Control Message Protocol	ICMP
2	Internet Group Management Protocol	IGMP
6	Transmission Control Protocol	TCP
17	User Datagram Protocol	UDP
41	IPv6 encapsulation	ENCAP
89	Open Shortest Path First	OSPF
132	Stream Control Transmission Protocol	SCTP

Source: Wikipedia

ICMP

- A **supporting protocol** for sending error messages and operational information
- Used by ping and traceroute tools
- ICMP header format:



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type								Code								Checksum															
Rest of Header																															

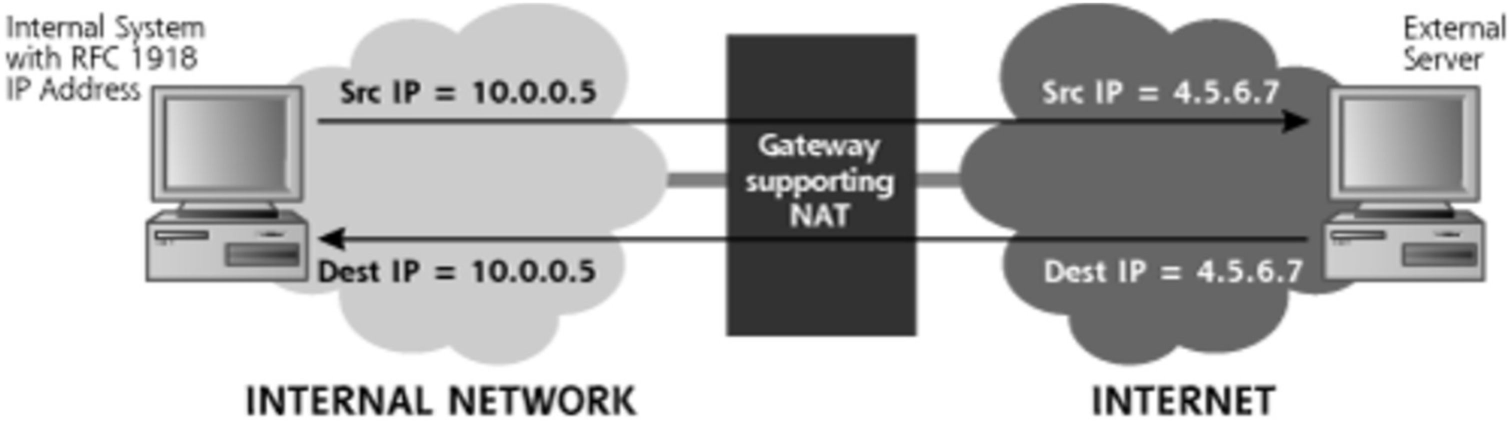
Source: Wikipedia

- Some control messages (with their **ICMP Types**):
  - Echo Reply (0), Destination Unreachable (3), Redirect Message (5), Echo Request (8), Time Exceeded (11), Parameter Problem: Bad IP header (12)

Special Network Devices

Network Address Translation (NAT)

- Necessary for private networks using private IP addresses to access the Internet
- Example:



Source: Skoudis & Liston, Counter Hack Reloaded

- Possible address mappings: to a single external IP address, 1-1 mapping, dynamic address mapping

Firewall

- Control flow of traffic *between* networks
- Different types of firewalls (based on network layer operations):
  - Traditional packet filters:
    - Check the following: source IP address, destination IP address, source TCP/UDP port, destination TCP/UDP port, TCP control bits, protocol in use, direction, interface
  - Stateful packet filters: keeps track of a state table
  - Proxy-based firewalls
- *Question: Differences with network-based IDS?*

Source: Wikipedia

Traceroute & Firewall: Extra Notes

- traceroute (UNIX):
  - Sends **UDP packets** by default
  - Can sends ICMP Echo Request (-I), or arbitrary protocol (-P)
- tracert (Windows):
  - sends **ICMP Echo Request** by default
- Firewalls **usually blocks** ICMP or unwelcome UDP!
- Other variants that use **TCP SYN** packets:
  - tcptraceroute (https://linux.die.net/man/1/tcptraceroute)
  - tctrace (http://manpages.ubuntu.com/manpages/cosmic/man1/tctrace.1.html)

Data Link Layer

Ethernet and 802.11

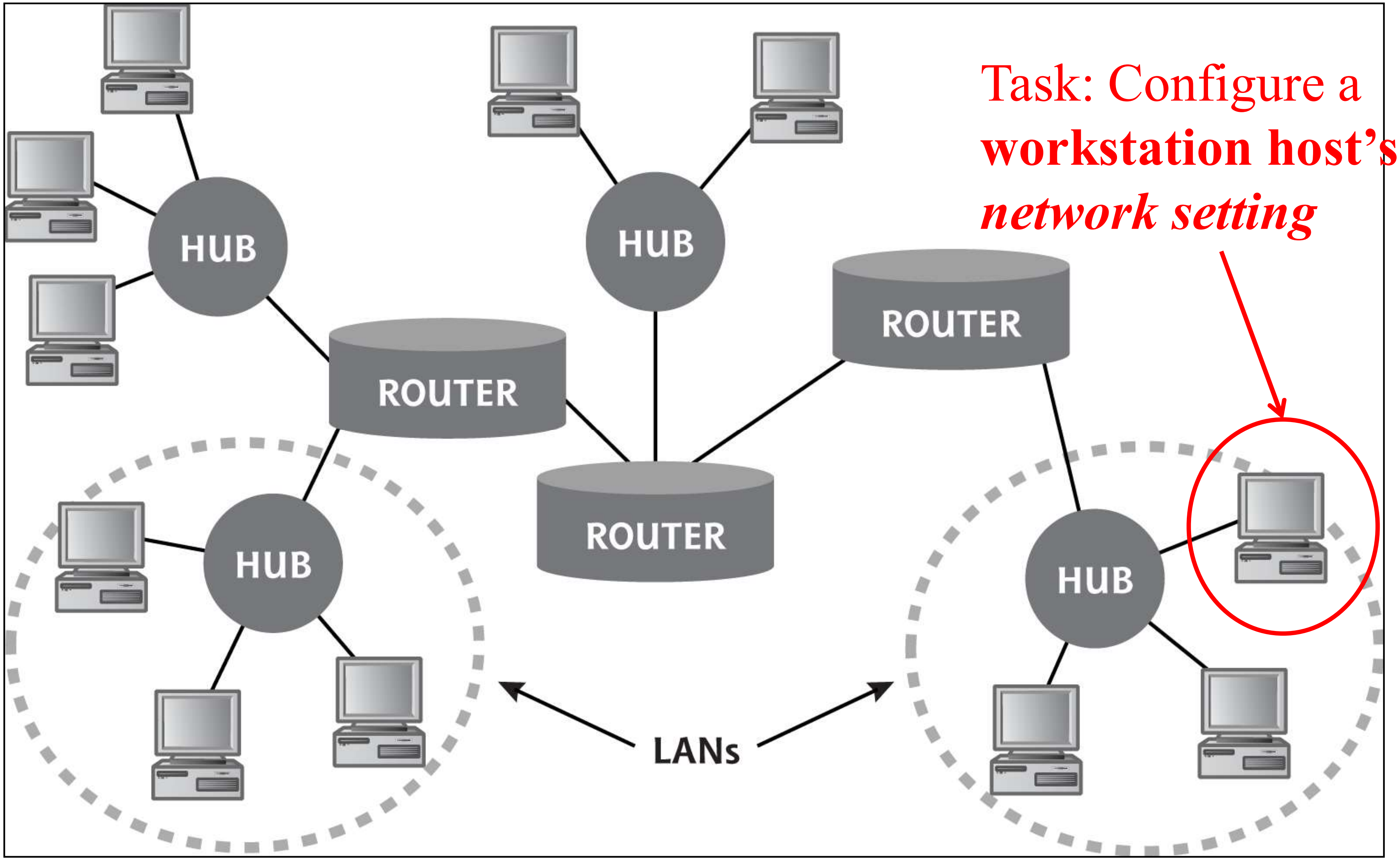
- Ethernet:
  - 48-bit MAC address
- Address Resolution Protocol (ARP):
  - Map logical IP address (layer 3) to physical MAC address (layer 2)
  - ARP Cache table for minimizing future ARP traffic
- Hubs vs switches:
  - Switches offer improved performance and better security
- 802.11: attacks on Ethernet are applicable too

Common Network Services

- telnet
- ssh
- ftp
- http
- r-commands: rlogin, rsh, rcp
- DNS
- NFS
- X Windows

Network Configuration: Linux Desktop

Setting up a Computer



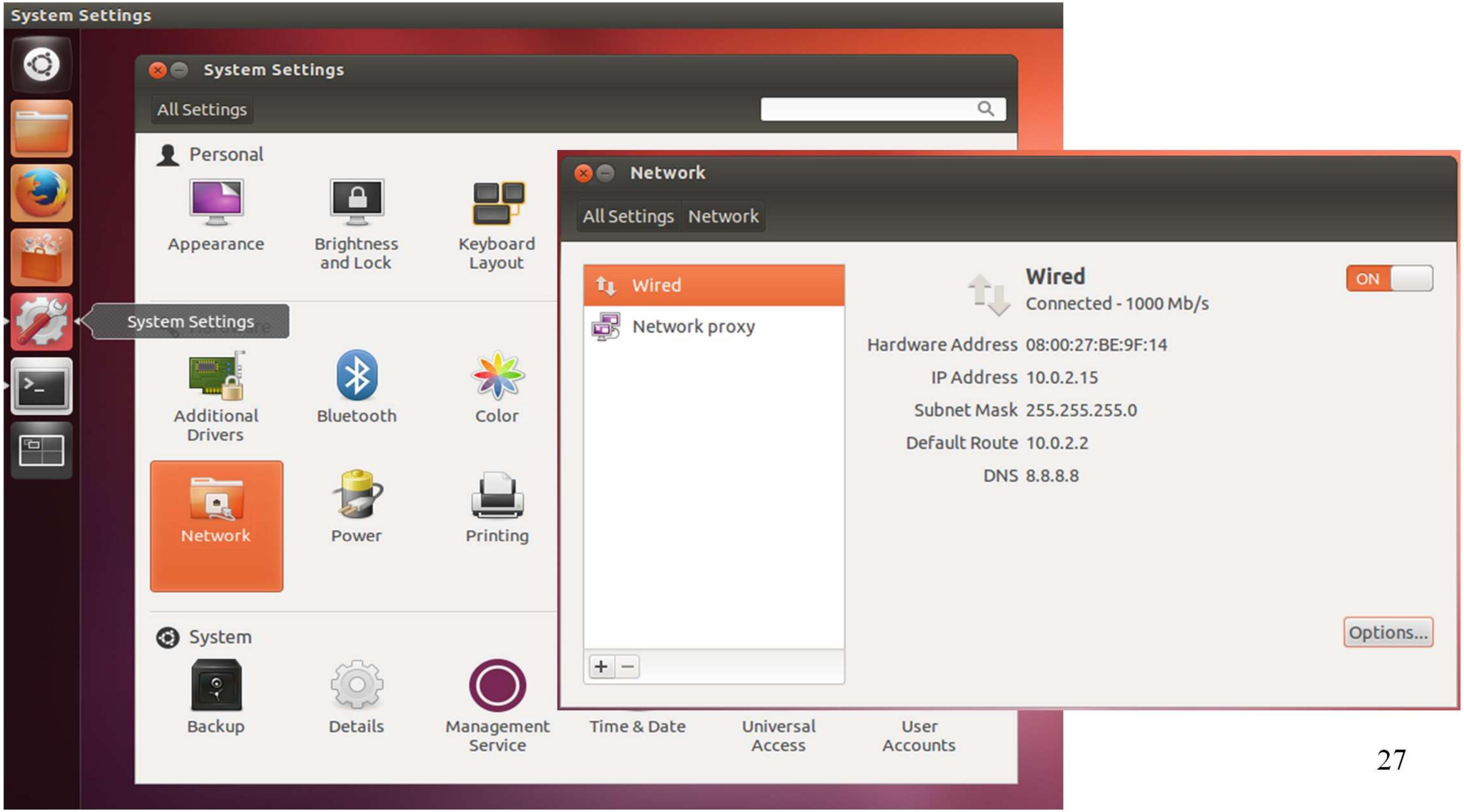
Task: Configure a **workstation host's** ***network setting***

Computer Network Configuration

- **Information** needed to connect a computer to the Internet:
    - IP Address
    - Network mask
    - Gateway
    - DNS server
- *How to **obtain** such information?*
    - Automatic setting through DHCP
    - Manual setting

Configuration in Ubuntu Linux

- “System Settings” → “Network”



Wired

Connected - 1000 Mb/s

ON

Hardware Address 08:00:27:BE:9F:14

IP Address 10.0.2.15

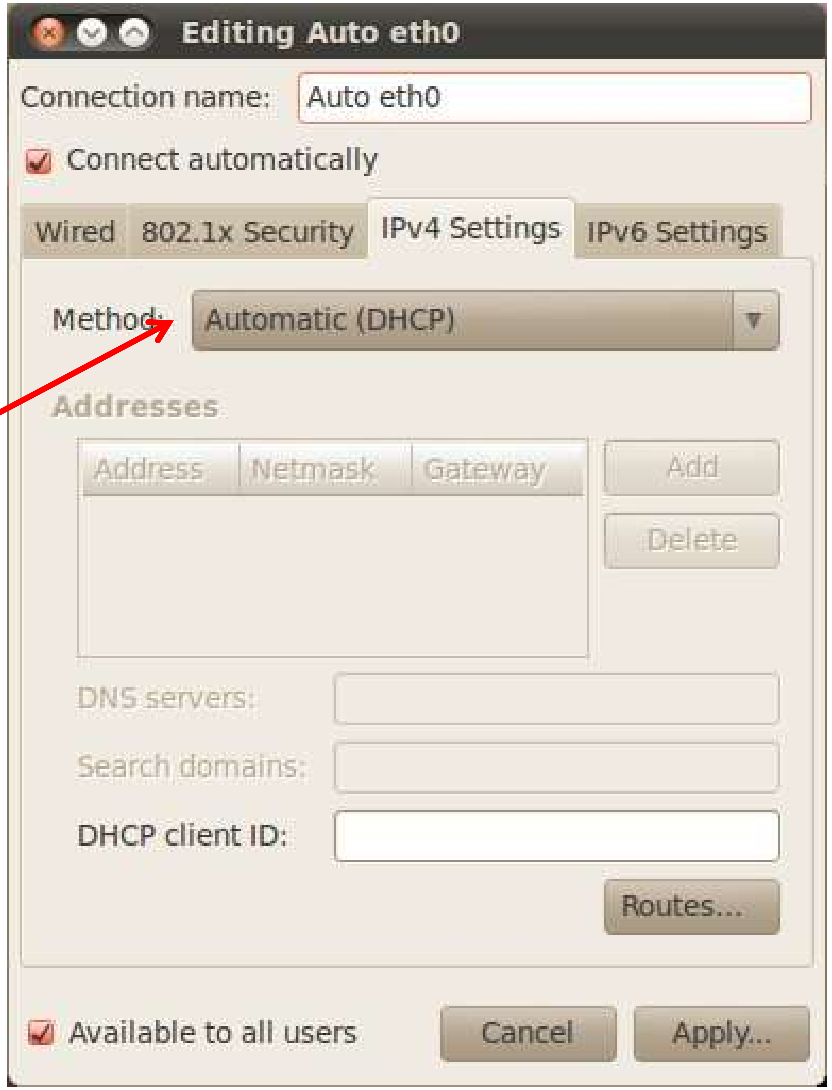
Subnet Mask 255.255.255.0

Default Route 10.0.2.2

DNS 8.8.8.8

Automatic Network Settings (DHCP)

- Select your network interface, and click the "**Options**" button
- Select "**IPv4 Settings**" tab
- Set method to "**Automatic (DHCP)**" in order to automatically obtain network settings from **DHCP server**



Connection name: Auto eth0

☑ Connect automatically

Method: Automatic (DHCP)

☑ Available to all users

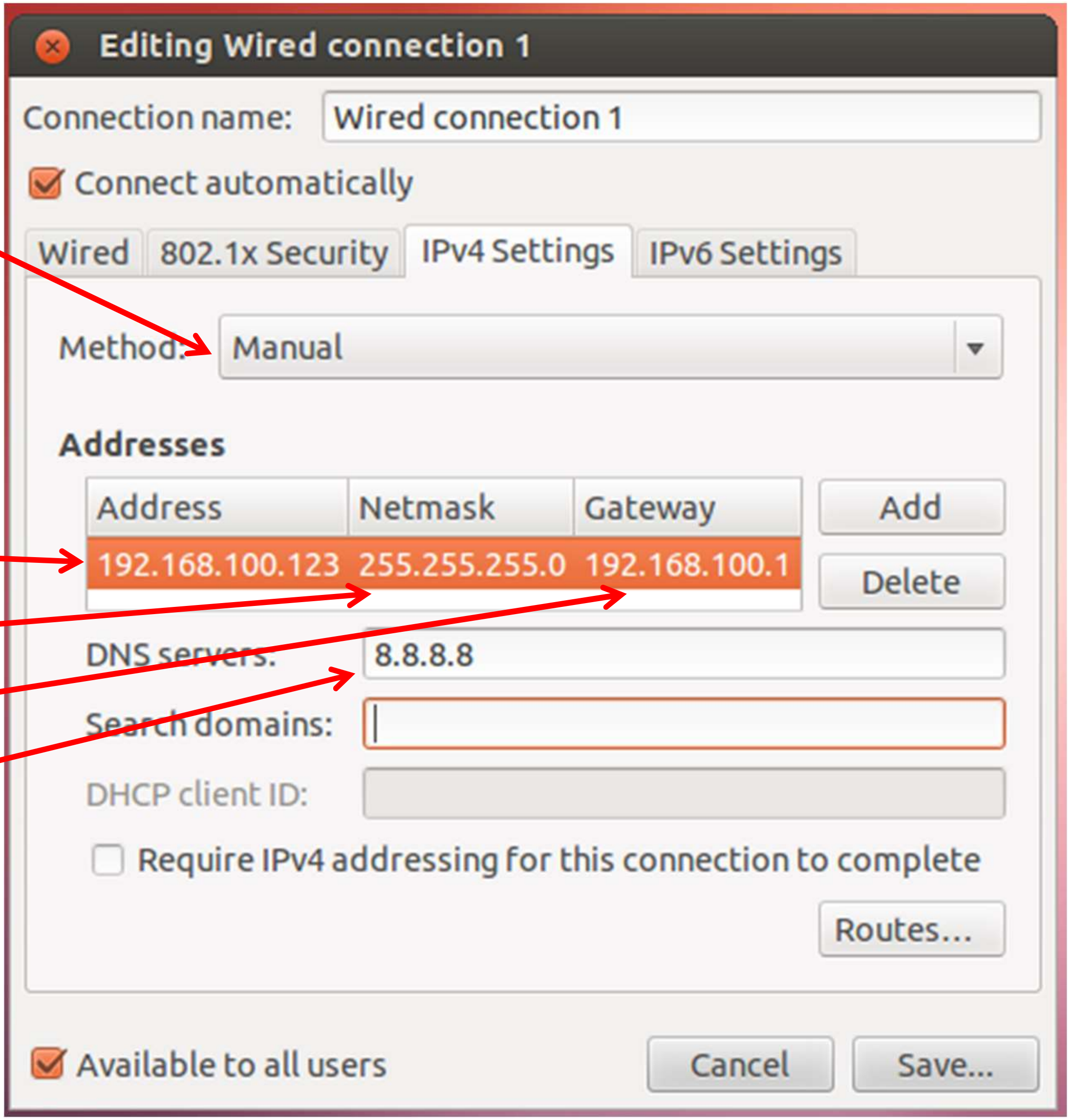
Consistent Network Device Naming

- A convention for naming **Ethernet adapters** in Linux
- Created ~2009 to replace the old eth*X* naming:
  - **Issues** on multihomed machines
  - NICs would be named based on **the order** in which they were found by the kernel as it booted
  - Removing existing or adding new interfaces?
- Device naming rules:
  - **Onboard** interfaces at firmware index nos: `eno`[1-*N*]
  - Interfaces at **PCI Express hotplug** slot nos: `ens`[1-*N*]
  - Adapters in the specified **PCI slot**, with slot index no on the adapter `enp`<*PCI-slot*>`s`<*card-index-no*>

Manual Network Settings

Set method to “**Manual**”

- IP Address
- Network mask
- Gateway
- DNS server



Editing Wired connection 1

Connection name: Wired connection 1

☑ Connect automatically

Method: Manual

Addresses

Address	Netmask	Gateway
192.168.100.123	255.255.255.0	192.168.100.1

DNS servers: 8.8.8.8

Search domains:

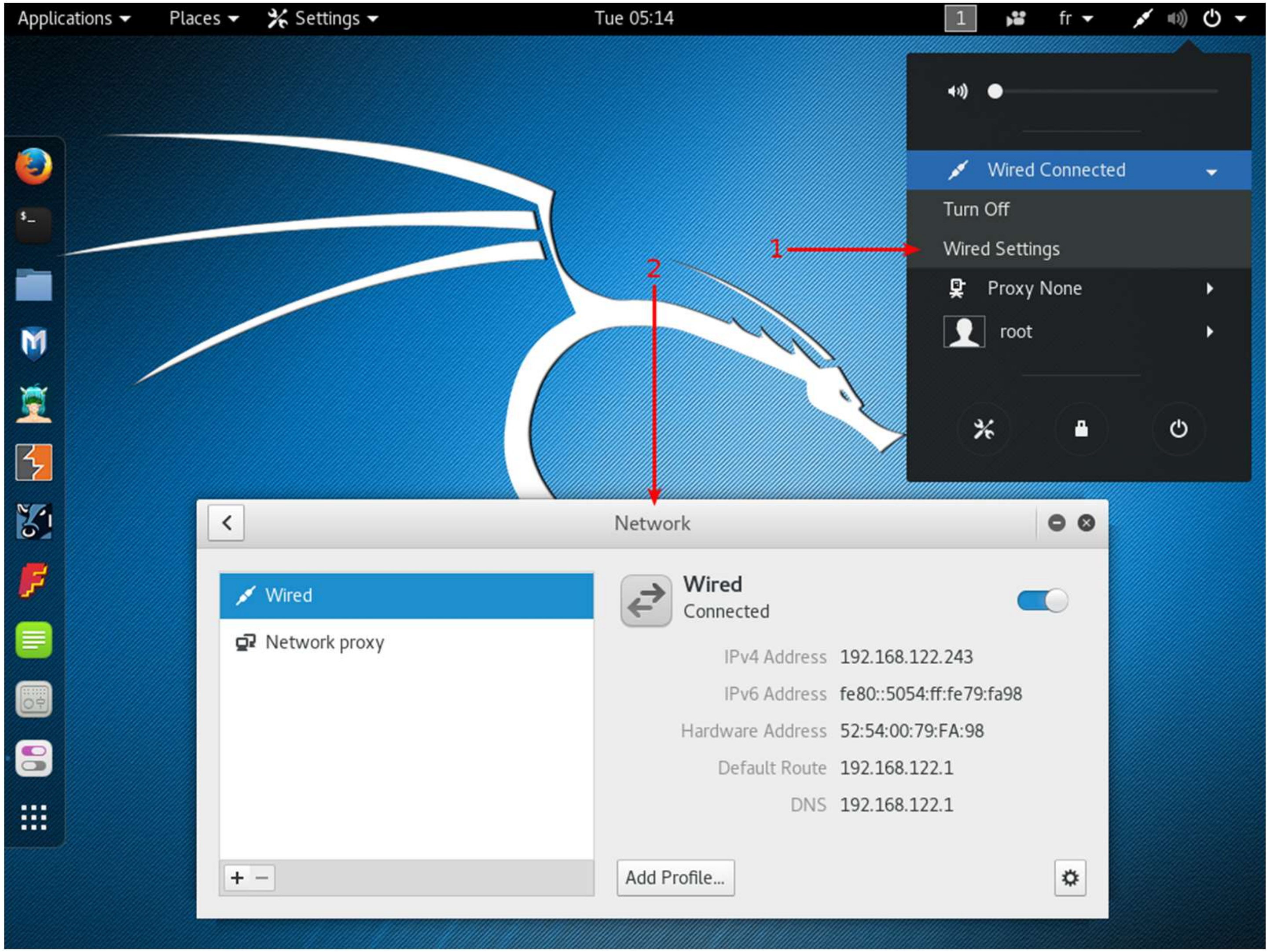
DHCP client ID:

□ Require IPv4 addressing for this connection to complete

☑ Available to all users

Configuration in Kali Linux

- *NetworkManager* setting interface:



Wired
Connected

IPv4 Address 192.168.122.243

IPv6 Address fe80::5054:ff:fe79:fa98

Hardware Address 52:54:00:79:FA:98

Default Route 192.168.122.1

DNS 192.168.122.1

Source: “Kali Linux Revealed”, Hertzog et al., 2017

Network Setting File and Commands

- **Manual** network setting steps:
  - `ifdown` *<network-device>*
  - Modify **`/etc/network/interfaces`**
  - `ifup` *<network-device>*
- Setting `/etc/network/interfaces` for a plain **DHCP configuration**:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

Network Setting File and Commands

- Setting `/etc/network/interfaces` for a **static IP** configuration:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.0.3
netmask 255.255.255.0
broadcast 192.168.0.255
network 192.168.0.0
gateway 192.168.0.1
```

Configuring Kali Linux: Services

- Managing services:
  - E.g. ssh:
    - `systemctl start ssh`
    - `systemctl enable ssh`
    - `systemctl reload ssh`
  - E.g. Apache:
    - `systemctl start apache2`
    - `a2enmod` *module*
    - `a2dismod` *module*

Test Your Configuration

- If your setting steps, you should be able to **connect** to the Internet:
- **Troubleshooting**: if your Internet connection doesn’t work, try to diagnose it:
  - Can you reach your gateway? (use `ping` command)
    - Note that ping may not work for various reasons
  - Can you reach your DNS server? (use `ping` command)
  - Can you resolve a domain name? (use `nslookup`)

Some Useful Commands

- Check and start/stop network interfaces using **`ifconfig`** :
  - **List** network interfaces:
    - **All** interfaces (**up and down**) whose drivers are loaded:

      ```
      $ ifconfig -a
      ```

    - All interfaces that are **up**:

      ```
      $ ifconfig
      ```

    - A particular interface (e.g. eth0):

      ```
      $ ifconfig eth0
      ```

  - **Start** and **stop** a network interface (e.g. eth0):

    ```
    $ ifconfig eth0 down
    $ ifconfig eth0 up
    ```

Some Useful Commands

- Newer **`ip`** command from **iproute2**:
  - **List** network interfaces:
    - **All** interfaces (up and down) whose drivers are loaded:

      ```
      $ ip addr show
      ```

    - A particular interface (e.g. eth0):

      ```
      $ ip addr show eth0
      ```

    - IPv4 or IPv6 addresses only:
    - `$ ip -4|-6 addr show`
  - **Start** and **stop** a network interface (e.g. eth0):

    ```
    $ ip link set eth0 down
    $ ip link set eth0 up
    ```

Linux Network Commands: Deprecated and New

- Old-style network utilities from net-tools (`ifconfig`, `route`, ...) are ***supposed*** to be replaced by iproute2:
  - `ifconfig` → `ip`
  - `route` → `ip`
  - `arp` → `ip`
  - `netstat` → `ss` (socket statistics)
- Sample command comparisons:
  - `route -n` vs `ip route show`
  - `route add default gw` *<gateway-IP-addr>* vs
    `ip route add default via` *<gateway-IP-addr>*

References: ip Command

- https://phoenixnap.com/kb/linux-ip-command-examples
- https://www.howtogeek.com/657911/how-to-use-the-ip-command-on-linux/