# Data-Link Layer

Course Code: **COE 3201** Course Title: **Data Communication**


**Dept. of Computer Engineering**
**Faculty of Engineering**

| | | | | | |
|---|---|---|---|---|---|
| **Lecture No:** | **12** | **Week No:** | **13** | **Semester:** | |
| **Lecturer:** | | | | | |

# Lecture Outline


1. Data-link layer Overview
2. Addressing mechanism
3. Link-layer addresses

# TCP/IP layers

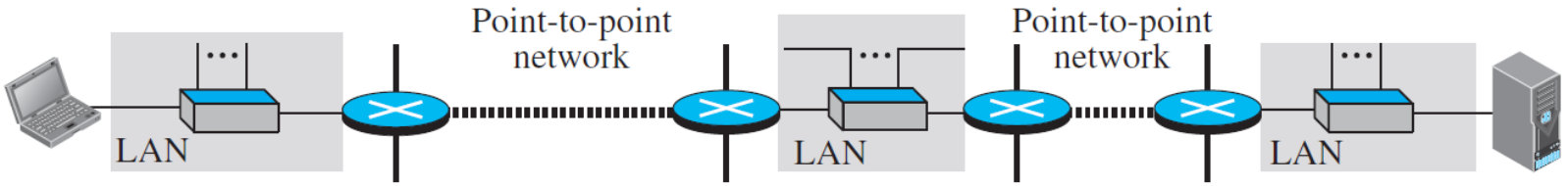
- Reviewing TCP/IP model, we note that:
  - Application layer protocols facilitate the operation of the program used by user.
  - Transport layer classifies which server and which client programs are supposed to receive which data.
  - Network layer lets networks to communicate with each other through routers.
  - The data-link layer interprets physical layer signals and gets data to nodes in the same network.
  - Physical layer handles cabling and sending signals and specifies how signals are sent over connections.

Next topic

Discussed so far

# Nodes and Links

- A data unit from one point in the Internet needs to pass through many networks (LANs and WANs) to reach another point.
- Theses LANs and WANs are connected by routers. Each router takes input from one network and sends output to another network.
- It is customary to refer to the two end hosts and the routers as **nodes** and the networks in between as **links**.




a. A small part of the Internet


b. Nodes and links

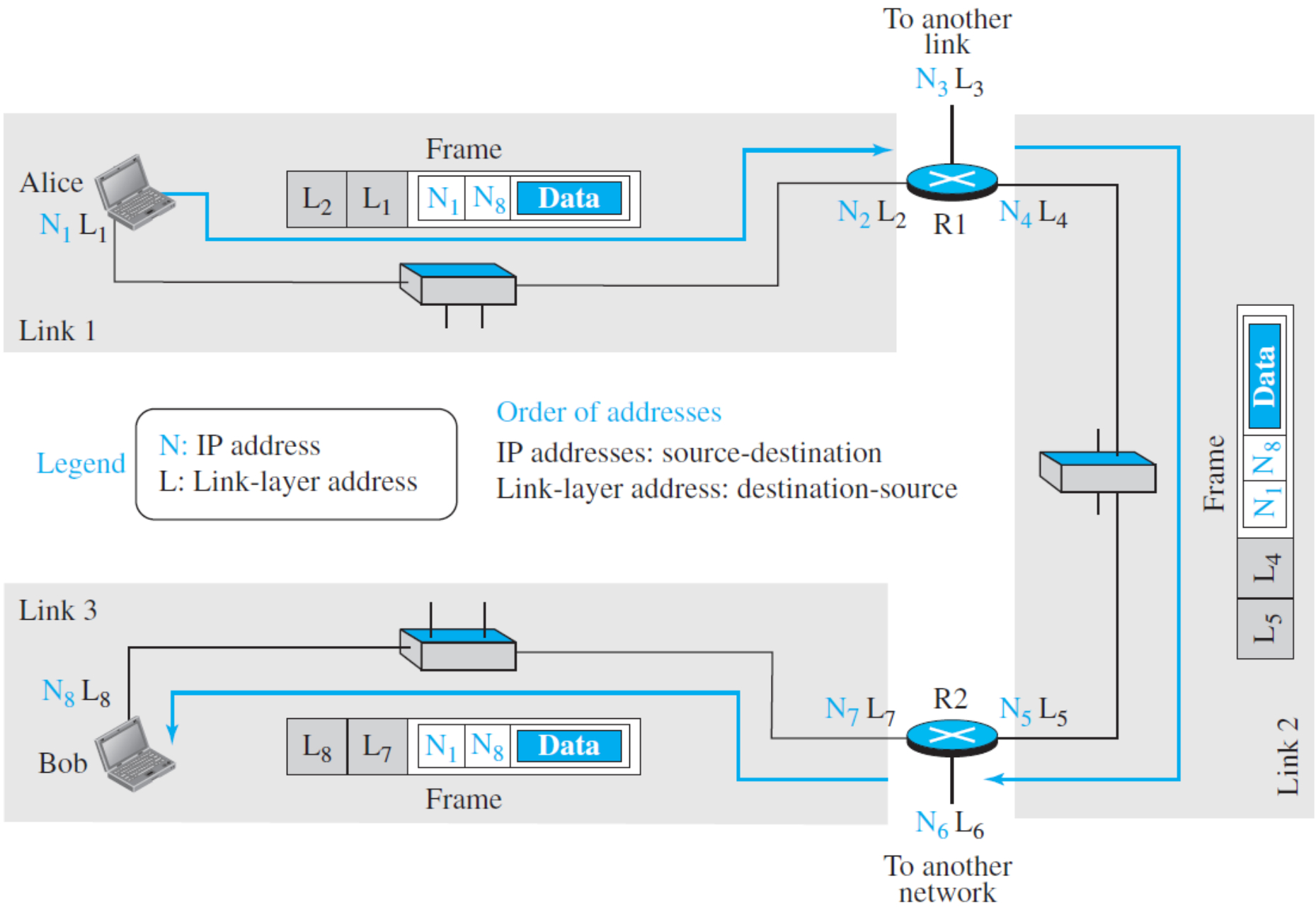
# Communication in the Data-Link Layer


- Communication at the data-link layer is **node-to-node**.
- When a packet is travelling in the Internet, the data-link layer of a node (host or router) is responsible for delivering a datagram to the next node in the path.
- The data-link layer of the source host needs only to encapsulate, the data-link layer of the destination host needs to decapsulate, but each intermediate node needs to both encapsulate and decapsulate.
- A packet at the data-link layer is normally called a **frame**.

# Addressing Mechanism for Data-Link Layer


- During communication over Internet, each unit of data may take a different path from the same source host to the same destination host.
- The source and destination **IP addresses** at the network layer define the two ends and should not be changed under normal circumstances, but they cannot define which links the datagram should pass through.
- For this purpose, a **data link layer address** is needed that gets data to **nodes in the same network**. Data link layer address is called a **MAC address** or Physical address.

**Figure 9.5** *IP addresses and link-layer addresses in a small internet*



# Link Layer Addresses

a) **Unicast:** Each host or each interface of a router is assigned a unicast address. Unicasting means one-to-one communication. **A3:34:45:11:92:F1** is an example of unicast link-layer address in common LAN or Ethernet. Here 48 bits are presented as 12 hexadecimal digits separated by colons.

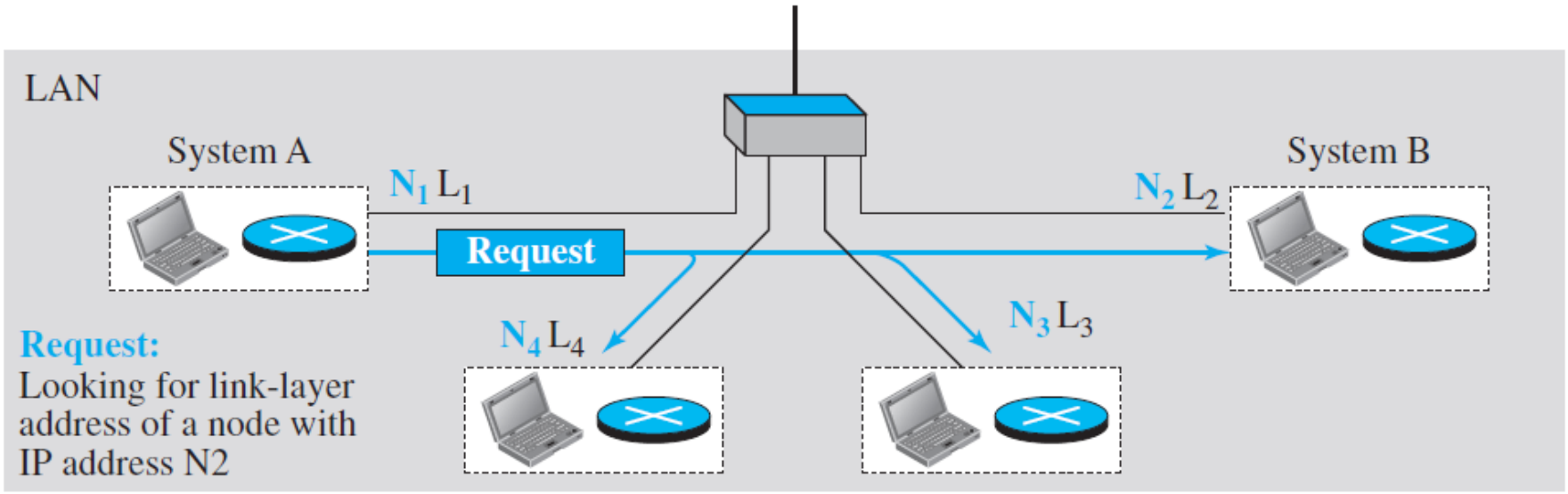
b) **Multicast:** Multicasting means one-to-many communication. However, the jurisdiction is local (inside the link). Follows similar pattern as unicast but the second digit needs to be an even number in hexadecimal, such as **A2:34:45:11:92:F1**

c) **Broadcast:** Broadcasting means one-to-all communication. A frame with a destination broadcast address is sent to all entities in the link. Here the 48 bits are all 1s, such as **FF:FF:FF:FF:FF:FF**

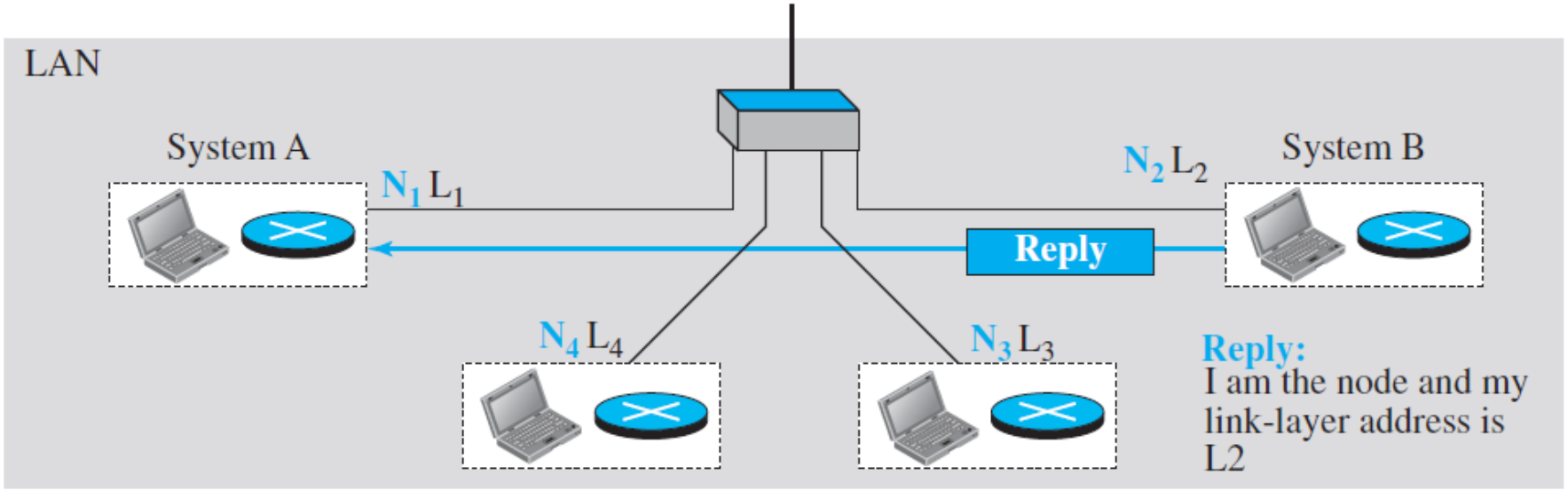
# Address Resolution Protocol (ARP)


- **Switches** are data-link layer devices that use MAC addresses to forward data to the right destination. (On the other hand, router is a network layer device.)
- **Address Resolution Protocol (ARP)** maps the IP addresses at the network layer to MAC addresses at the data-link layer in an ARP table. This protocol helps a packet at the network layer find the link-layer address of the next node.
- If MAC address of the next node is not known to the sender (but IP address is known), sender transmits an ARP broadcast to all nodes in the same network, and only the receiver node responds and shares its MAC address.

**Figure 9.7** *ARP operation*




a. ARP request is broadcast



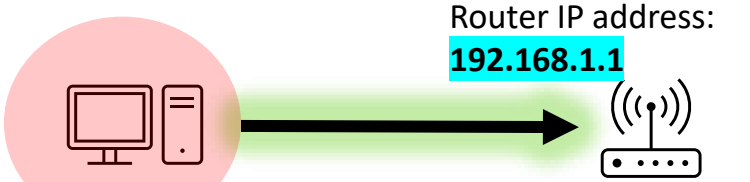

b. ARP reply is unicast

# Example


- Next, a detailed example of communication between a personal computer and a web server **connected to the same network** is considered in terms of all TCP/IP layers.
- Note that **MAC addresses** of data-link layer keep changing as a packet travels across links, but the source and destination **IP addresses** of network layer stays the same.
- Additionally, ports are transport layer information that allow data to be directed to or from the correct service.
- Encapsulation and decapsulation happens in each node.

**Computers in same network**

**Sending from my computer → Router**



Router IP address:
**192.168.1.1**

My computer IP address:
**192.168.1.21**
My computer port:
6457(random)
My computer MAC address:
**b7:27:7a:ad:e7:6a**

Router MAC address:
**17:a5:d0:a0:4c:c7**

Web server IP address:
**192.168.1.31**
Web server port: 8080
Web server MAC address:
**af:e3:9b:60:c1:22**

LLC: TCP or UDP
Source MAC:
b7:27:7a:ad:e7:6a
Destination MAC:
17:a5:d0:a0:4c:c7

Source IP: 192.168.1.21
Destination IP:
192.168.1.31
TTL, Checksum

Source port: 6457
Destination port: 8080
Checksum

Connection request from browser application

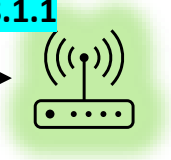
Checksum

**Computers in same network**

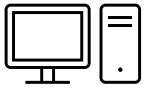
**Sending from my computer → Router**

My computer IP address:
**192.168.1.21**
My computer port:
6457(random)
My computer MAC address:
**b7:27:7a:ad:e7:6a**

Router IP address:
**192.168.1.1**



Router MAC address:
**17:a5:d0:a0:4c:c7**



Web server IP address:
**192.168.1.31**
Web server port: 8080
Web server MAC address:
**af:e3:9b:60:c1:22**

LLC: TCP or UDP
Source MAC:
b7:27:7a:ad:e7:6a
Destination MAC:
17:a5:d0:a0:4c:c7

Source IP: 192.168.1.21
Destination IP:
192.168.1.31
TTL, Checksum

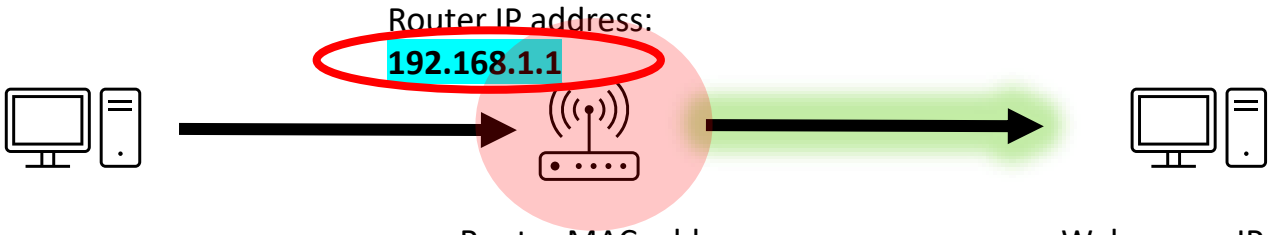
Source port: 6457
Destination port: 8080
Checksum

Connection request from browser application

Checksum

**Computers in same network**

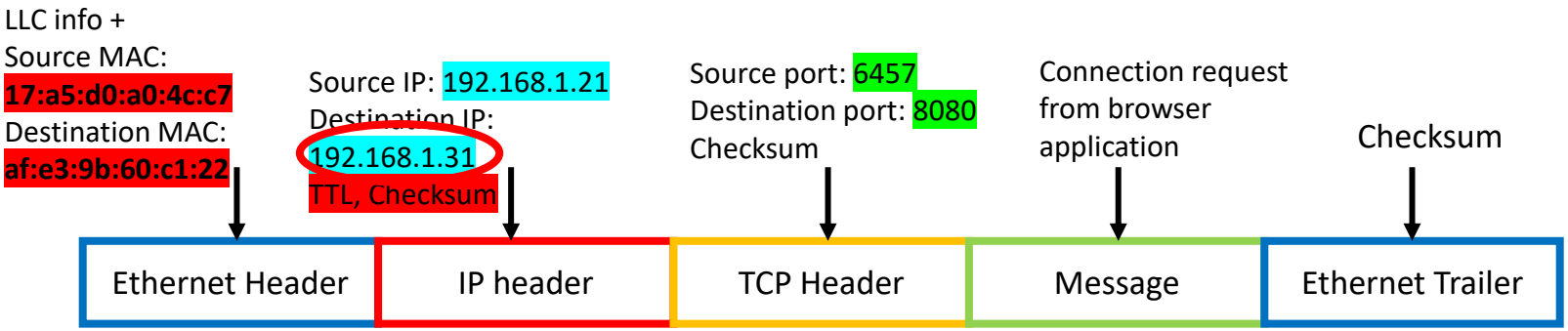
**Sending from Router → Web Server**




My computer IP address:
**192.168.1.21**
My computer port:
6457(random)
My computer MAC address:
**b7:27:7a:ad:e7:6a**

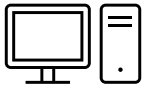
Router MAC address:
**17:a5:d0:a0:4c:c7**

Web server IP address:
**192.168.1.31**
Web server port: 8080
Web server MAC address:
**af:e3:9b:60:c1:22**



**Computers in same network**

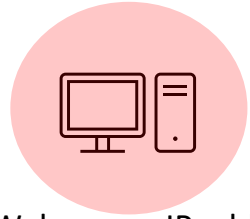
**Once Web Server has received**



My computer IP address:
**192.168.1.21**
My computer port:
6457(random)
My computer MAC address:
**b7:27:7a:ad:e7:6a**

Router IP address:
**192.168.1.1**

Router MAC address:
**17:a5:d0:a0:4c:c7**



Web server IP address:
**192.168.1.31**
Web server port: 8080
Web server MAC address:
**af:e3:9b:60:c1:22**

LLC info +
Source MAC:
**17:a5:d0:a0:4c:c7**
Destination MAC:
**af:e3:9b:60:c1:22**

Source IP: 192.168.1.21
Destination IP:
192.168.1.31
TTL, Checksum

Source port: 6457
Destination port: 8080
Checksum

Connection request from browser application

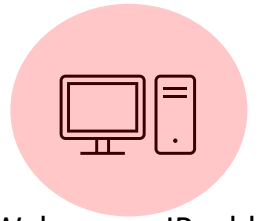
Checksum

**Computers in same network**

**Web Server -> Router**

Router IP address:
**192.168.1.1**





My computer IP address:
**192.168.1.21**
My computer port:
6457(random)
My computer MAC address:
**b7:27:7a:ad:e7:6a**

Router MAC address:
**17:a5:d0:a0:4c:c7**

Web server IP address:
**192.168.1.31**
Web server port: 8080
Web server MAC address:
**af:e3:9b:60:c1:22**

LLC info +
Source MAC:
**af:e3:9b:60:c1:22**
Destination MAC:
**17:a5:d0:a0:4c:c7**

Source IP: 192.168.1.31
Destination IP:
192.168.1.21
TTL, Checksum

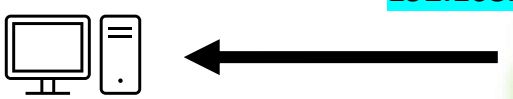
Source port: 8080
Destination port: 6457
Checksum

Data from web page on server

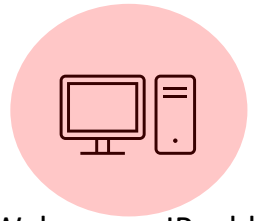
Checksum

**Computers in same network**

**Router->My Computer**



Router IP address:
**192.168.1.1**



My computer IP address:
**192.168.1.21**
My computer port:
6457(random)
My computer MAC address:
**b7:27:7a:ad:e7:6a**

Router MAC address:
**17:a5:d0:a0:4c:c7**

Web server IP address:
**192.168.1.31**
Web server port: 8080
Web server MAC address:
**af:e3:9b:60:c1:22**

LLC info +
Source MAC:
**af:e3:9b:60:c1:22**
Destination MAC:
**17:a5:d0:a0:4c:c7**

Source IP: 192.168.1.31
Destination IP:
192.168.1.21
TTL, Checksum

Source port: 8080
Destination port: 6457
Checksum

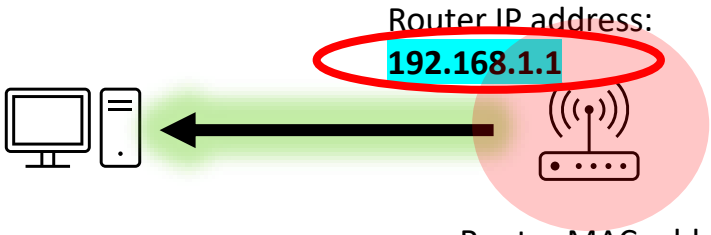
Data from web page on server

Checksum

| Ethernet Header | IP header | TCP Header | Message | Ethernet Trailer |
|---|---|---|---|---|

**Computers in same network**

**Router->My Computer**

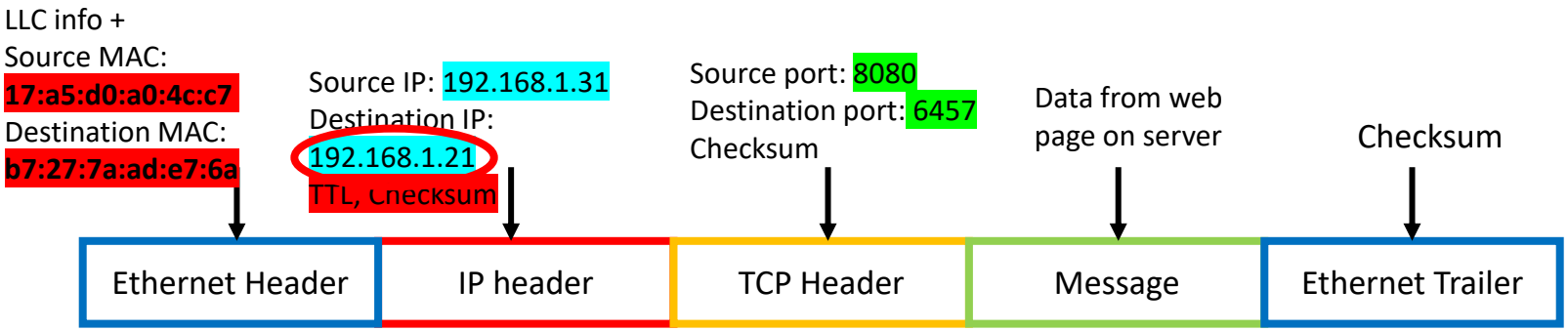


Router IP address:
192.168.1.1

My computer IP address:
192.168.1.21
My computer port:
6457(random)
My computer MAC address:
b7:27:7a:ad:e7:6a

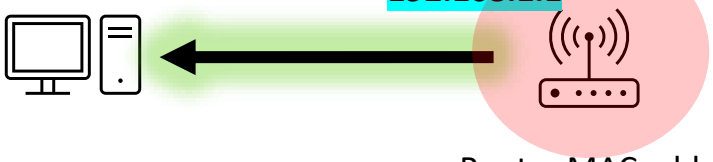
Router MAC address:
17:a5:d0:a0:4c:c7

Web server IP address:
192.168.1.31
Web server port: 8080
Web server MAC address:
af:e3:9b:60:c1:22



**Computers in same network**

**Once My Computer has received**



Router IP address:
192.168.1.1

My computer IP address:
192.168.1.21
My computer port:
6457(random)
My computer MAC address:
b7:27:7a:ad:e7:6a

Router MAC address:
17:a5:d0:a0:4c:c7

Web server IP address:
192.168.1.31
Web server port: 8080
Web server MAC address:
af:e3:9b:60:c1:22

LLC info +
Source MAC:
17:a5:d0:a0:4c:c7
Destination MAC:
b7:27:7a:ad:e7:6a

Source IP: 192.168.1.31
Destination IP:
192.168.1.21
TTL, Checksum

Source port: 8080
Destination port: 6457
Checksum

Data from web page on server

Checksum

**Computers in same network**

**Once My Computer has received**

Router IP address:
**192.168.1.1**

My computer IP address:
**192.168.1.21**
My computer port:
6457(random)
My computer MAC address:
**b7:27:7a:ad:e7:6a**

Router MAC address:
**17:a5:d0:a0:4c:c7**

Web server IP address:
**192.168.1.31**
Web server port: 8080
Web server MAC address:
**af:e3:9b:60:c1:22**