

Module 2

Sockets/ Part 3

Prepared by Dr. Manal Mizher

Lecture 4

جامعة الجميع الذكية

تسعى لتكون الكيان الأول الذي يتبنى مبدأ المجتمع التعليمي
التعاوني ويقدم العلم مقابل العمل لتنمية وبناء المجتمعات.

Network programming
Course ID: CYB615

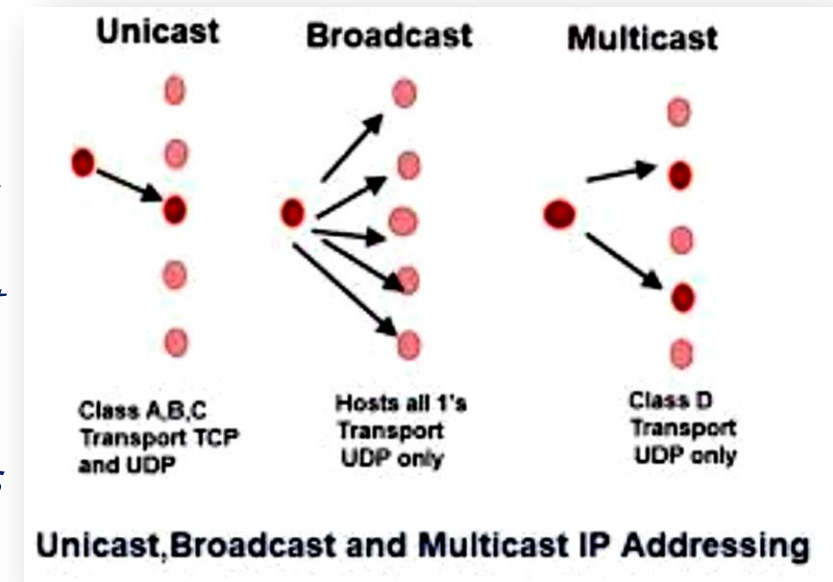


جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

(Sockets/Part 3_ Computer Networks Protocols(UDP_TCP_ICMP))

1. **Various topics:** (UDP broadcast, Using TCP Streams Like Files)
2. **Handling errors:** (Socket Exceptions, Missed Errors, Errors with File-like Objects)
3. **ICMP protocol (Definition, Packet, Header, Messages types)**
4. **ICMP in Python and raw sockets**
 1. Ping
 2. Broadcast Ping
 3. Multi-Ping
 4. Traceroute
 5. Verbose Ping
 6. Verbose traceroute

- UDP broadcast is a technique that allows sending UDP datagram from a single source to all computers in a LAN/entire subnet. (*sent to a special address called the Broadcast address*).
- *IPv6 doesn't even have a concept of broadcast (multicast is used, instead).*



A network mask, or **“netmask”**, is used to divide an IP address into a network address and a host address. When you set up a network, the network mask must be common to all network interfaces on that network. The default network masks are 255.0.0.0 for class A, 255.255.0.0 for class B, and 255.255.255.0 for class C networks.

	IP Address	netmask
Class A	16.1.1.1 network host	255.0.0.0
Class B	172.16.1.1 network host	255.255.0.0
Class C	221.138.62.1 network host	255.255.255.0

How to Find the Broadcast Address?

$$\begin{array}{r} 172.16.20.19 \quad 255.255.240.0 \\ \begin{array}{l} 16 + 16 \\ = 32 - 1 \\ = 31 \end{array} \quad \begin{array}{r} 172.16.\boxed{20}.19 \\ 255.255.240.0 \\ \hline 172.16.31.255 \end{array} \quad \begin{array}{r} 256 \\ - 240 \\ \hline 16 \end{array} \\ \text{Broadcast Address} = 172.16.31.255 \end{array}$$

- A broadcast IP address is only assigned once in each network. It is always the **last IP address of the subnet**.
- The network address and the broadcast address are not used as computer IP addresses.
- In a network, the first address field is reserved for the network and the last for the broadcast address.

```
def client(network, port):  
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)  
    sock.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)  
    text = 'Broadcast datagram!'  
    sock.sendto(text.encode('ascii'), (network, port))
```

- File objects can `read()` and `write()`, sockets can `send()` and `recv()`, and no kind of object can do both.
- Sometimes you will want to treat a socket like a normal Python file object—often because you want to pass it into code like that of the many Python modules such as `pickle`, `json`, and `zlib` that can read and write data directly from a file. For this purpose, Python provides a `makefile()` method on every socket that returns a Python file object that is really calling `recv()` and `send()` behind the scenes.

Module 2

Using TCP Streams Like Files

```
conn, addr = sock.accept()
print('socket accepted, got connection object')
sockFile = conn.makefile()
```

(Server side)

```
message = sockFile.readline()
print('received and read by socket: ' + str(message))
conn.close()
```

(Client side)

```
sock.connect((HOST, PORT))
print('socket connected')

sockFile = sock.makefile(mode='w')

message = 'Hi this is my msg Number = ' + str(myCounter) + ' ***' + '\r\n'
print('*** ' + message)

sockFile.write(message)
```

- In Python, the socket code raises exceptions when network errors occur.
- Virtually every function call that touches the network in any way can and does raise exceptions for various reasons, for example: servers being down, connections dropped, and so on.

1. Socket Exceptions

1. `socket.error` for general I/O and communication problems
2. `socket.gaierror` for errors looking up address information
3. `socket.herror` for other addressing errors (*This exception is thrown when there is a problem with `gethostbyname()` and `gethostbyaddr()`*)
4. `socket.timeout` for handling timeouts that occur after `settimeout()` has been called on a socket (Python 2.3 and higher)

2. Missed Errors

- There are certain situations in which communication problems could occur but no exception would be raised because no error was passed back from the operating system.
- One such problem could occur if the remote server drops the connection between the time the client connects and the time it writes out its request. In this case, the later call to `recv()` will receive no data (since the server closed its connection) and the program will terminate successfully.

3. Errors with File-like Objects

- It's possible to use the `makefile()` function to get a file-like object from a socket.
- This file-like object actually makes calls to the real socket, so the exceptions raised by the file-like object are the same as the ones raised by the socket's own `send()` and `recv()` functions.

```
fd = s.makefile('rw', 2048)
print('sleeping :')
time.sleep(5)
print('Continuing ...')

try:
    fd.write("GET %s HTTP/1.1" % filename)
    print('fd has this info: ', fd)
except socket.error as e:
    print('Error sending data:', e)
    sys.exit(1)
```

```
while 1:
    try:
        buf = fd.read(2048)
    except socket.error as e:
        print('Error receiving data:', e)
        sys.exit(1)
    if not len(buf):
        break
    sys.stdout.write(buf)
```

1. Definition of ICMP protocol
2. Redirect message
3. ICMP (Packet/ Header)
4. ICMP messages types
5. ICMP Protocol in Python
6. Raw Sockets
7. ICMP Functions (Ping, Traceroute, ...)

- **The Internet Control Message Protocol (ICMP)** is a supporting protocol in the Internet protocol suite.
- It is used by network devices, including routers, to send error messages and operational information indicating success or failure when communicating with another IP address, for example:
An error is indicated when a requested service is not available or that a host or router could not be reached.
- ICMP is not typically used to exchange data between systems, nor is it regularly employed by end-user network applications (with the exception of some diagnostic tools like ping and traceroute).
- ICMP for IPv4 is defined in RFC 792.
- ICMPv6, defined by RFC 4443, is used with IPv6.

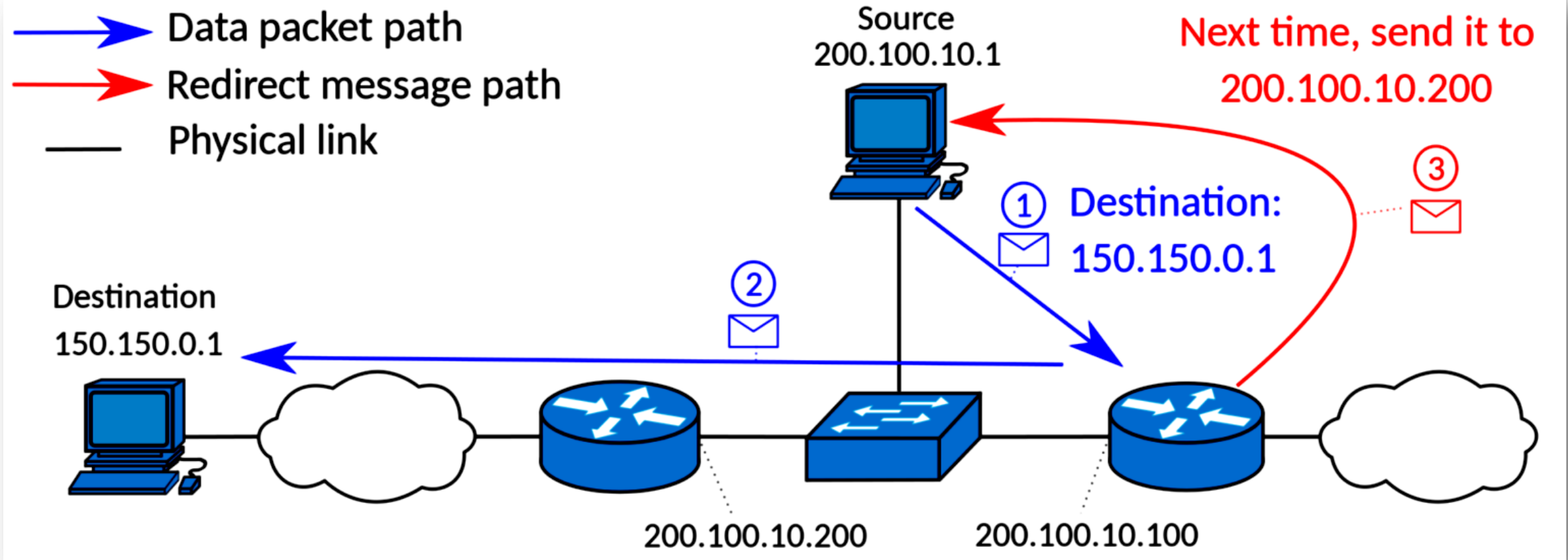
Definition: ICMP Protocol

- ICMP is a network-layer protocol. There is no TCP or UDP port number associated with ICMP packets as these numbers are associated with the transport layer above (It does not have source and destination port numbers because it was designed to communicate network-layer information between hosts and routers, not between application layer processes).
- ICMP messages are typically used for diagnostic or control purposes or generated in response to errors in IP operations (as specified in RFC 1122). ICMP errors are directed to the source IP address of the originating packet.

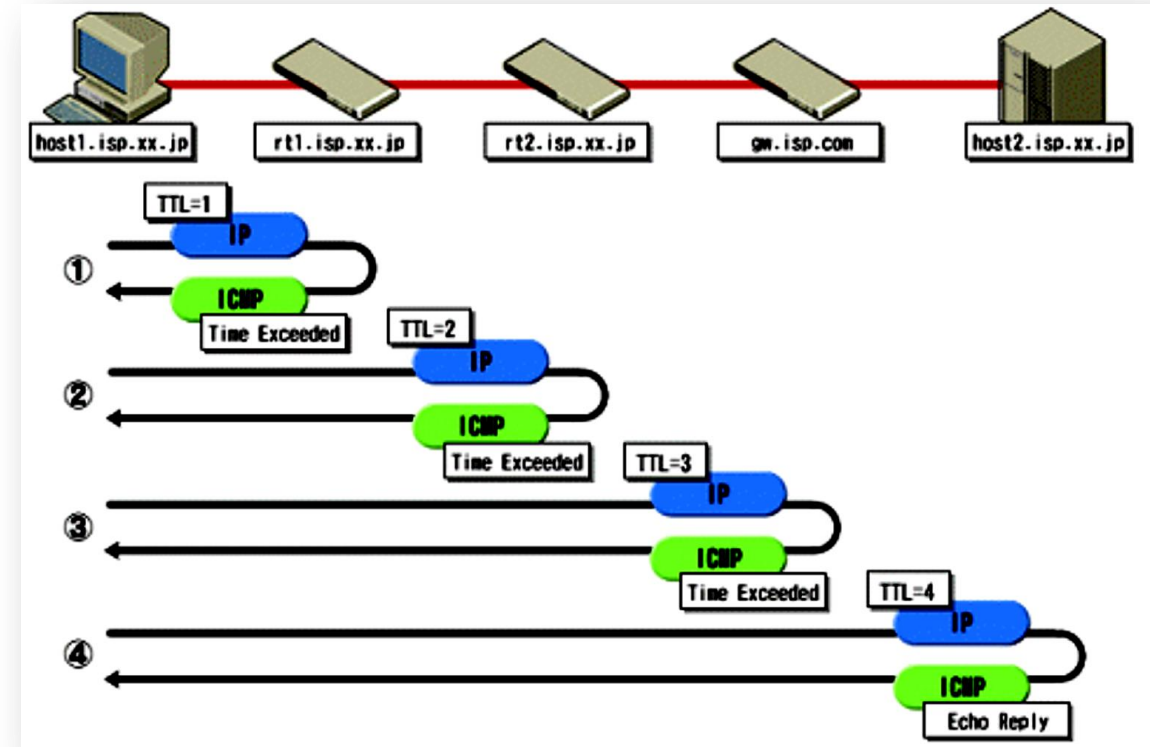
- ICMP uses the basic support of IP as if it were a higher-level protocol, however, ICMP is actually an integral part of IP. Although ICMP messages are contained within standard IP packets, ICMP messages are usually processed as a special case, distinguished from normal IP processing.
- In many cases, it is necessary to inspect the contents of the ICMP message and deliver the appropriate error message to the application responsible for transmitting the IP packet that prompted the ICMP message to be sent.

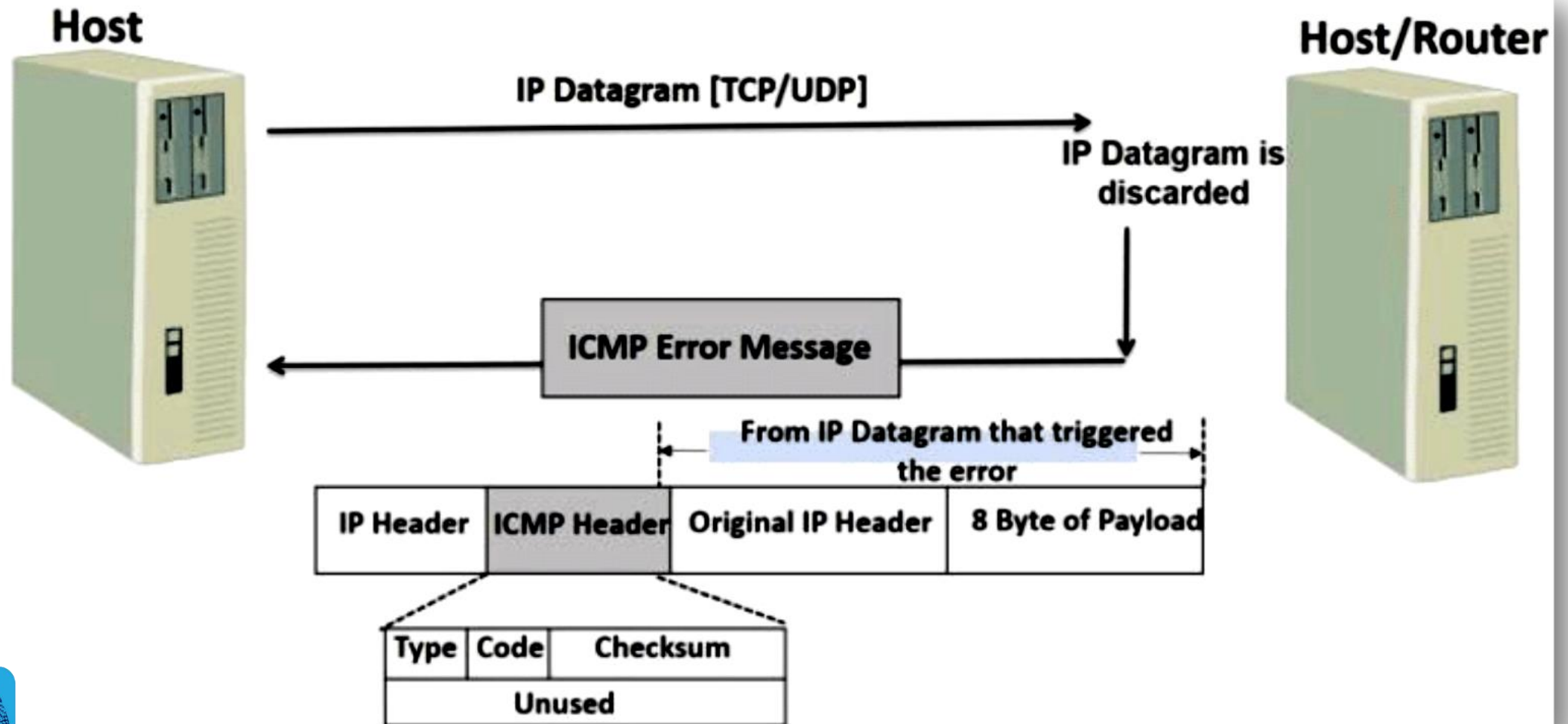
Module 2

An example of how an ICMPv4 redirect message works

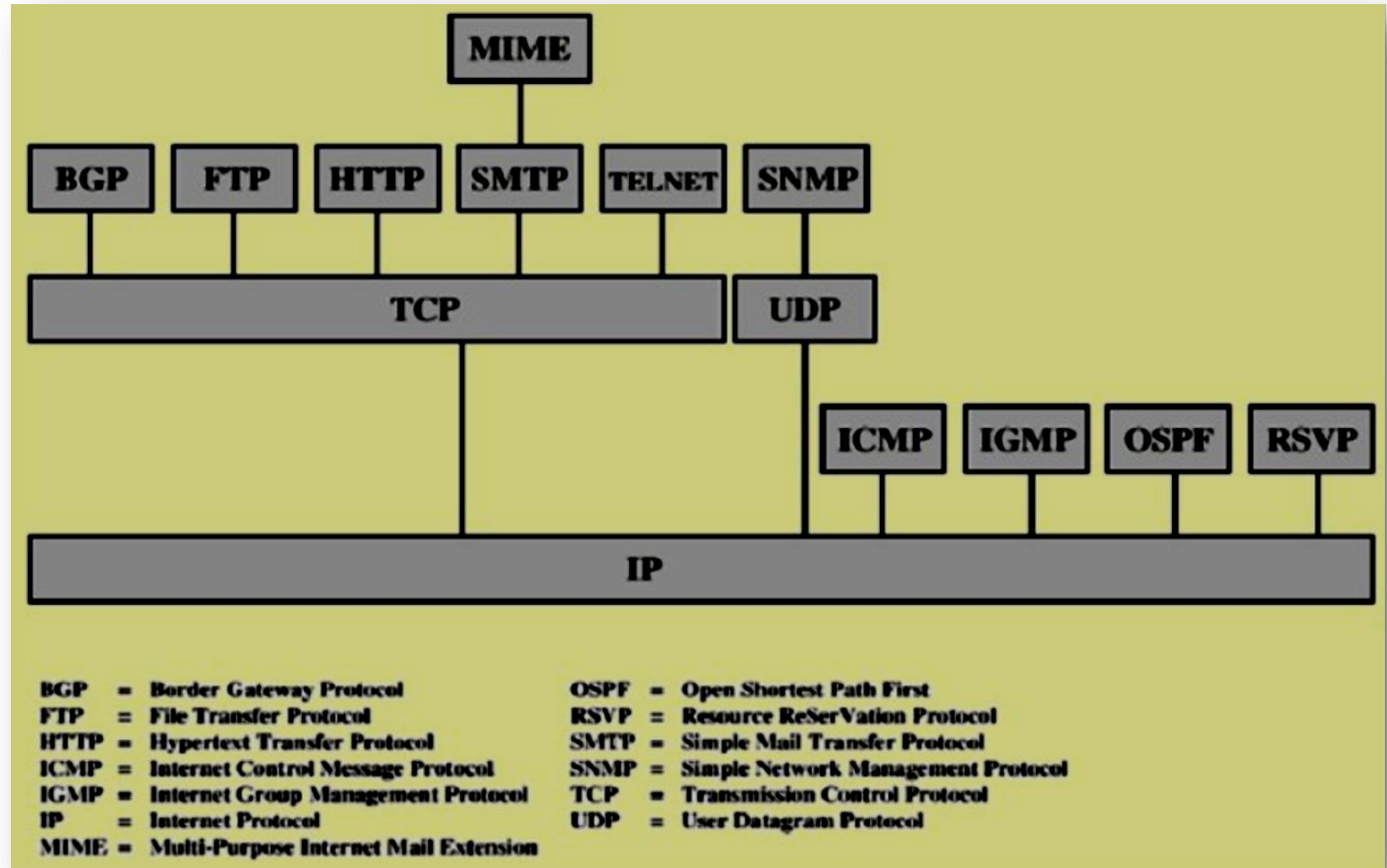


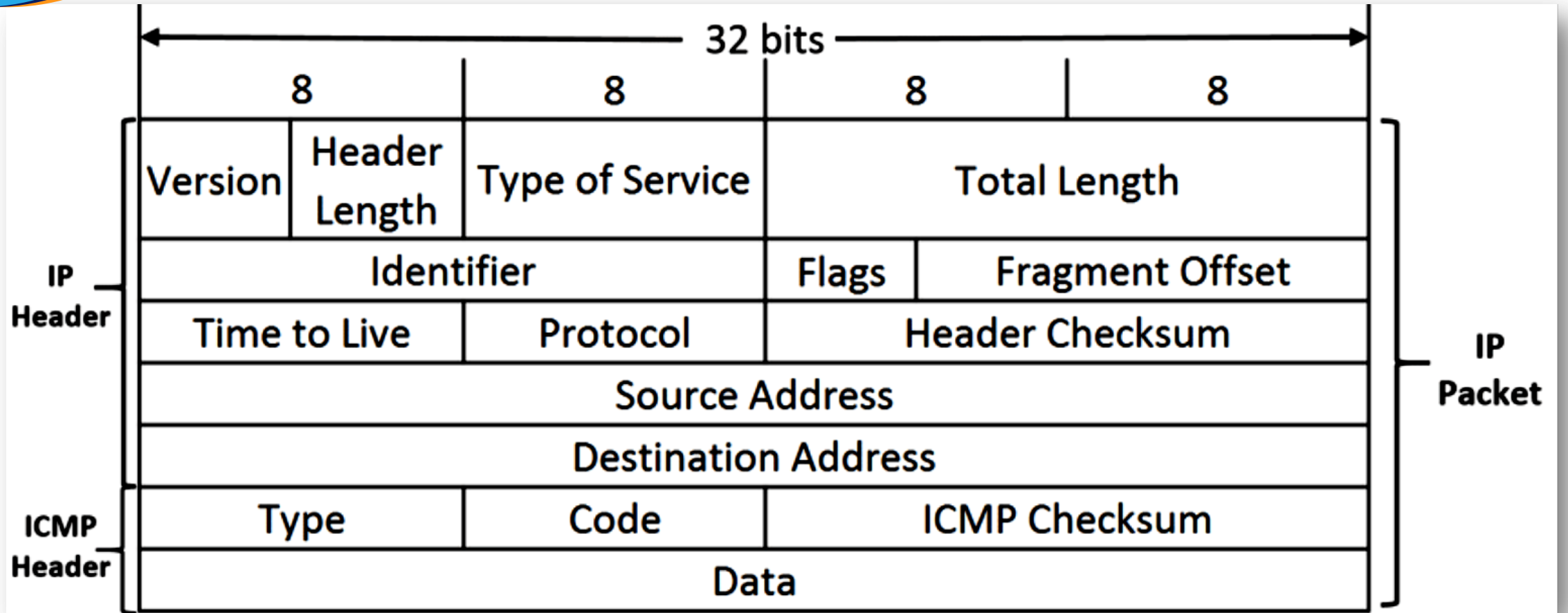
- For example, every device (such as an intermediate router) forwarding an IP datagram first decrements the time to live (TTL) field in the IP header by one. If the resulting TTL is 0, the packet is discarded and an ICMP time exceeded in transit message is sent to the datagram's source address.
- Many commonly used network utilities are based on ICMP messages. The traceroute command can be implemented by transmitting IP datagrams with specially set IP TTL header fields, and looking for ICMP time exceeded in transit and ***Destination unreachable*** messages generated in response. The related ping utility is implemented using the ICMP echo request and echo reply messages.





- ICMP messages:
Are encapsulated in IP packets so most people would say that it's a layer 4 protocol like UDP or TCP.
- However, since ICMP is a vital part of the IP protocol it is typically considered a layer 3 protocol.





IP header	ICMP header	ICMP payload size	MTU (1500)
20 bytes	8 bytes	1472 bytes (maximum)	$20 + 8 + 1472 = 1500$

- According to MTU the size of the ICMP packet cannot be greater than 1500 bytes.
- ICMP is actually a user of the IP protocol--in other words, ICMP messages must be encapsulated within IP packets. However, ICMP is implemented as part of the IP layer. So ICMP processing can be viewed as occurring parallel to, or as part of, IP processing.

Example of checksum calculation

8	0	0
1	9	
TEST		
8 & 0	→	00001000 00000000
0	→	00000000 00000000
1	→	00000000 00000001
9	→	00000000 00001001
T & E	→	01010100 01000101
S & T	→	01010011 01010100
Sum	→	10101111 10100011
Checksum	→	01010000 01011100

- ICMP uses unicast routing for error detection and notification messages.

- What is the difference between ICMP and ping?

Ping is a tool commonly used to find the status of a device on a network. Ping is based on the ICMP protocol. When a Ping process request is sent out as an ICMP echo to the target device, it replies with an ICMP echo reply if the device is available.

Error-reporting messages

The error-reporting message means that the router encounters a problem when it processes an IP packet then it reports a message.

Query messages

The query messages are those messages that help the host to get the specific information of another host. For example, suppose there are a client and a server, and the client wants to know whether the server is live or not, then it sends the ICMP message to the server.

ICMP Message Category	Type	Message
Error / Reporting Messages	3	Destination Unreachable
	4	Source Quence
	5	Redirection
	11	Time Exceeded
	12	Parameter Problem
Information / Query Messages	0	Echo Request
	8	Echo Reply
	9	Router Advertisement
	10	Router Solicitation
	13	Timestamp Request
	14	Timestamp Reply
	15	Information Request
	16	Information Reply
	17	Timestamp Request
	18	Timestamp Reply



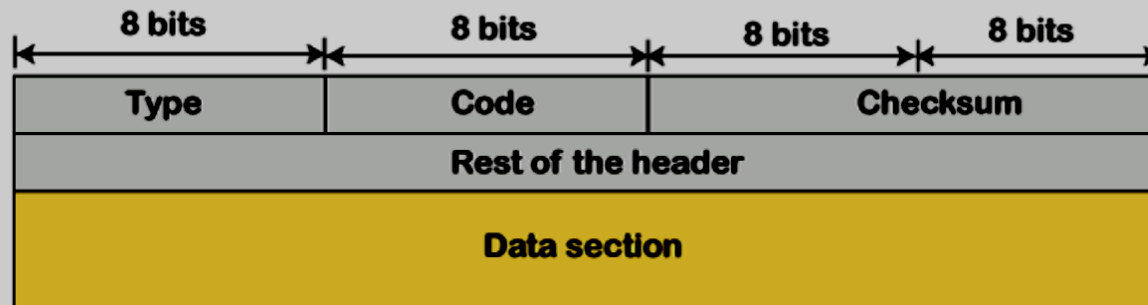
ICMP Messages

Type	Query/Error (Error Type)	Code	Description
0	Query	0	Echo reply
3	Error: Destination unreachable	0	Network unreachable
		1	Host unreachable
		2	Protocol unreachable
		3	Port unreachable
		4	Fragmentation needed, but the Don't Fragment bit has been set
		5	Source route failed
		6	Destination network unknown
		7	Destination host unknown
		8	Source host isolated (obsolete)
3	Error: Destination unreachable	9	Destination network administratively prohibited
		10	Destination host administratively prohibited
		11	Network unreachable for TOS
		12	Host unreachable for TOS
		13	Communication administratively prohibited by filtering
		14	Host precedence violation
		15	Precedence cutoff in effect
4	Error	0	Source quench
5	Error: Redirect	0	Redirect for network
		1	Redirect for host
		2	Redirect for TOS and network
		3	Redirect for TOS and host
8	Query	0	Echo request
9	Query	0	Router advertisement
10	Query	0	Router solicitation
11	Error: Time exceeded	0	TTL equals 0 during transit
		1	TTL equals 0 during reassembly
12	Error: Parameter problem	0	IP header bad
		1	Required option missing

ICMP Message Format

The message format has two things; one is a category that tells us which type of message it is. If the message is of error type, the error message contains the type and the code. The type defines the type of message while the code defines the subtype of the message.

The ICMP message contains the following fields:



- **Type:** It is an 8-bit field. It defines the ICMP message type. The values range from 0 to 127 are defined for ICMPv6, and the values from 128 to 255 are the informational messages.
- **Code:** It is an 8-bit field that defines the subtype of the ICMP message
- **Checksum:** It is a 16-bit field to detect whether the error exists in the message or not.

Error Message : Type 3

◦ Destination unreachable

The destination unreachable error occurs when the packet does not reach the destination. Suppose the sender sends the message, but the message does not reach the destination, then the intermediate router reports to the sender that the destination is unreachable.

Type: 3	Code: 0 to 15	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

- Type: the number 3 specifies that the destination is unreachable.
- Code (0 to 15): It is a 4-bit number which identifies whether the message comes from some intermediate router or the destination itself.
 - Sometimes the destination does not want to process the request, so it sends the destination unreachable message to the source. A router does not detect all the problems that prevent the delivery of a packet.

ICMP Query Messages

The ICMP Query message is used for error handling or debugging the internet. This message is commonly used to ping a message.

Echo-request and echo-reply message

Type 8: Echo request
Type 0: Echo reply

A **router** or a host can send an echo-request message. It is used to ping a message to another host that "Are you alive". If the other host is alive, then it sends the echo-reply message. An echo-reply message is sent by the router or the host that receives an echo-request message.

Key points of Query messages

1. The echo-request message and echo-reply message can be used by the network managers to check the operation of the IP protocol. Suppose two hosts, i.e., A and B, exist, and A wants to communicate with host B. The A host can communicate to host B if the link is not broken between A and B, and B is still alive.
2. The echo-request message and echo-reply message check the host's reachability, and it can be done by invoking the ping command.

Timestamp-request and timestamp-reply message

The timestamp-request and timestamp-reply messages are also a type of query messages. Suppose the computer A wants to know the time on computer B, so it sends the timestamp-request message to computer B. The computer B responds with a timestamp-reply message.

Message format of timestamp-request and timestamp-reply

Type 13: request

Type 14: reply

Type: 13 or 14	Code: 0	Checksum
Identifier		Sequence number
Original timestamp		
Receive timestamp		
Transmit timestamp		

Key points related to timestamp-request and timestamp-reply message

- It can be used to calculate the round-trip time between the source and the destination, even if the clocks are not synchronized.
- It can also be used to synchronize the clocks in two different machines if the exact transit time is known.

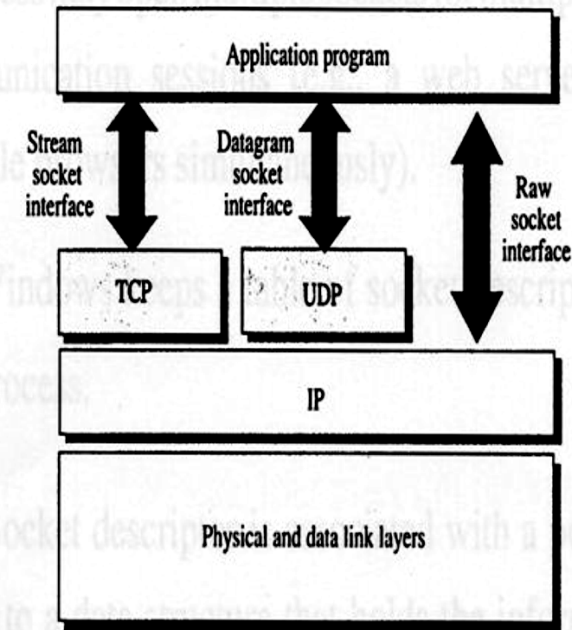
If the sender knows the exact transit time, then it can synchronize the clock. The sender asks the time on the receiver's clock, and then it adds the time and propagation delay. Suppose the time is 1:00 clock and propagation delay is 100 ms, then time would be 1:00 clock plus 100 ms.

1. ICMP always reports error messages to the original source.
2. No ICMP error message will be generated in response to a datagram carrying an ICMP error message.
3. No ICMP error message will be generated for a fragmented datagram that is not the first fragment.
4. No ICMP error message will be generated for a datagram having a multicast address.
5. No ICMP error message will be generated for a datagram having a special address such:
 - a) 127.0.0.0
 - b) 0.0.0.0



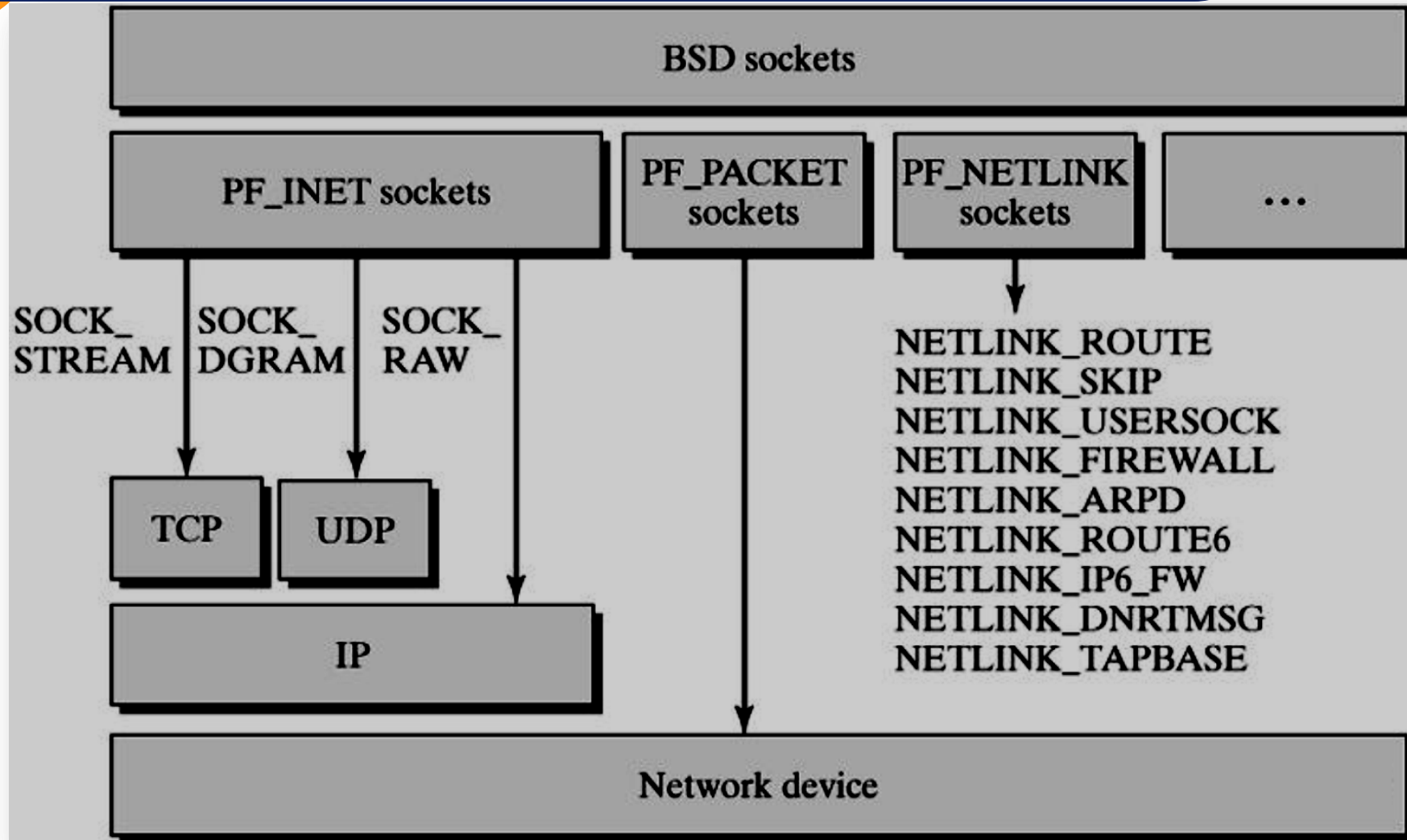
Types of Sockets

- A **stream socket** uses TCP for connection-oriented communication. It is also called a TCP socket.
- A **datagram socket** uses UDP for connectionless communication. It is also called a UDP socket.
- A **raw socket** uses IP directly. It is used in such applications as in **ICMP** protocol

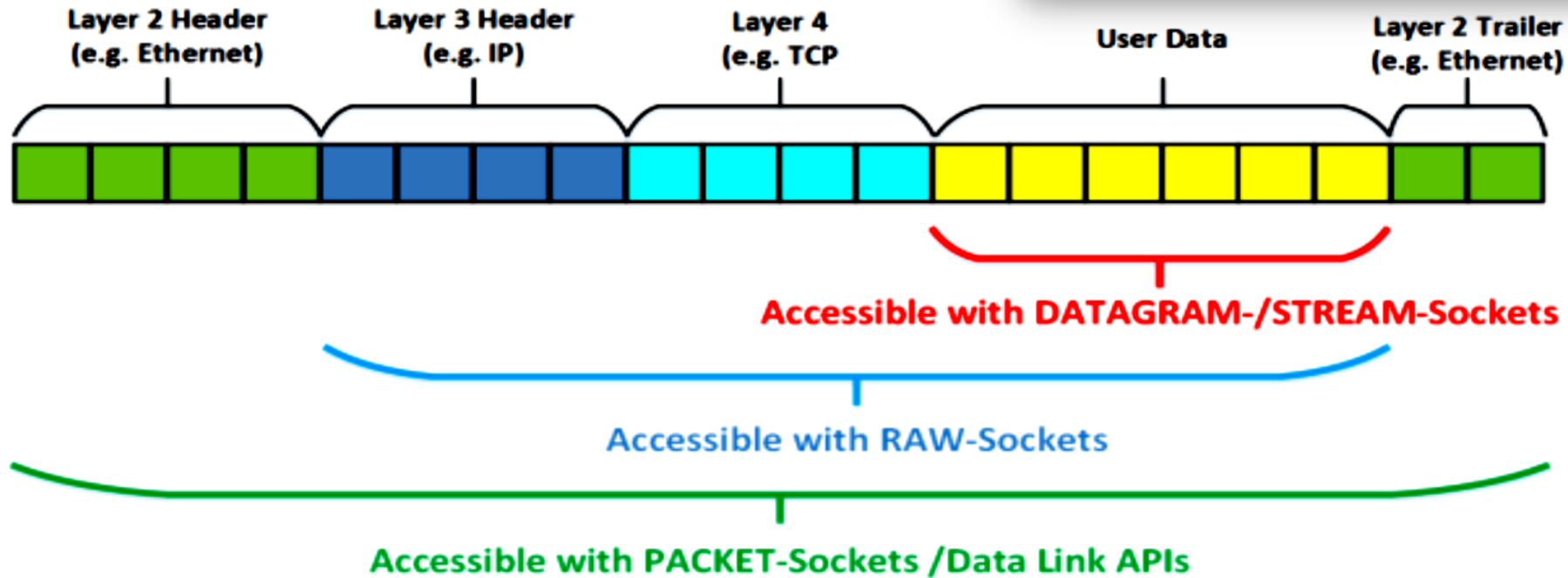


- Raw IP packet manipulation

ICMP & Raw sockets



Overview over the layers and access possibilities



- The biggest problem with RAW-sockets (also the PACKET-sockets) is that there is no uniform API for using RAW-sockets under different operating systems.

- Raw socket is a type of network socket which provides an **express tunnel between an application and an external source**. Neither the operating system nor any other program on the computer has the ability to interfere with a raw connection. (**Need root privilege/ Administrator**).
- The primary advantage to using raw socket network handling is that it cuts out the middle man. Since the operating system doesn't handle the data specifically:
 1. Reduce overhead on the network.
 2. Save central processing unit (CPU) cycles.
 3. Decrease stress on the system hardware.

- Raw socket network interfaces do have a downside:

Hackers commonly use raw socket connections to stage transmission control protocol (TCP) attacks on a network. During a TCP attack, a hacker sends a forged bit of data onto the network through a raw socket connection. This forged data contains a reset signal for the TCP connection, which in turn interrupts and crashes the current network connections on the computer.

- For this reason, some operating systems have withdrawn support for raw sockets.
- Software companies can restrict users' ability to use raw sockets.
- With modern processors coming in dual-, quad-, and even six-core varieties, the chances of regular network socket connections lagging the computer are negligible. For this reason, unless there is a specific justification for using a raw socket connection over a standard network socket, the risk posed by hackers and TCP attacks on the network outweighs any advantages.

- icmplib 3.0.3
- The power to form ICMP packets and do ping and traceroute. <https://pypi.org/project/icmplib/>
- icmplib is a brand new and modern implementation of the ICMP protocol in Python.
- Latest version : Released: Feb 6, 2022
- *Since icmplib 3, Python 3.7 or later is required to use the library.*
- *If you are using Python 3.6 and you cannot update it, you can still install icmplib 2.*

- **Import basic functions**

from icmplib import ping, multiping, traceroute, resolve

- **Import asynchronous functions**

from icmplib import async_ping, async_multiping, async_resolve

- **Import sockets (advanced)**

from icmplib import ICMPv4Socket, ICMPv6Socket, AsyncSocket, ICMPRequest, ICMPReply

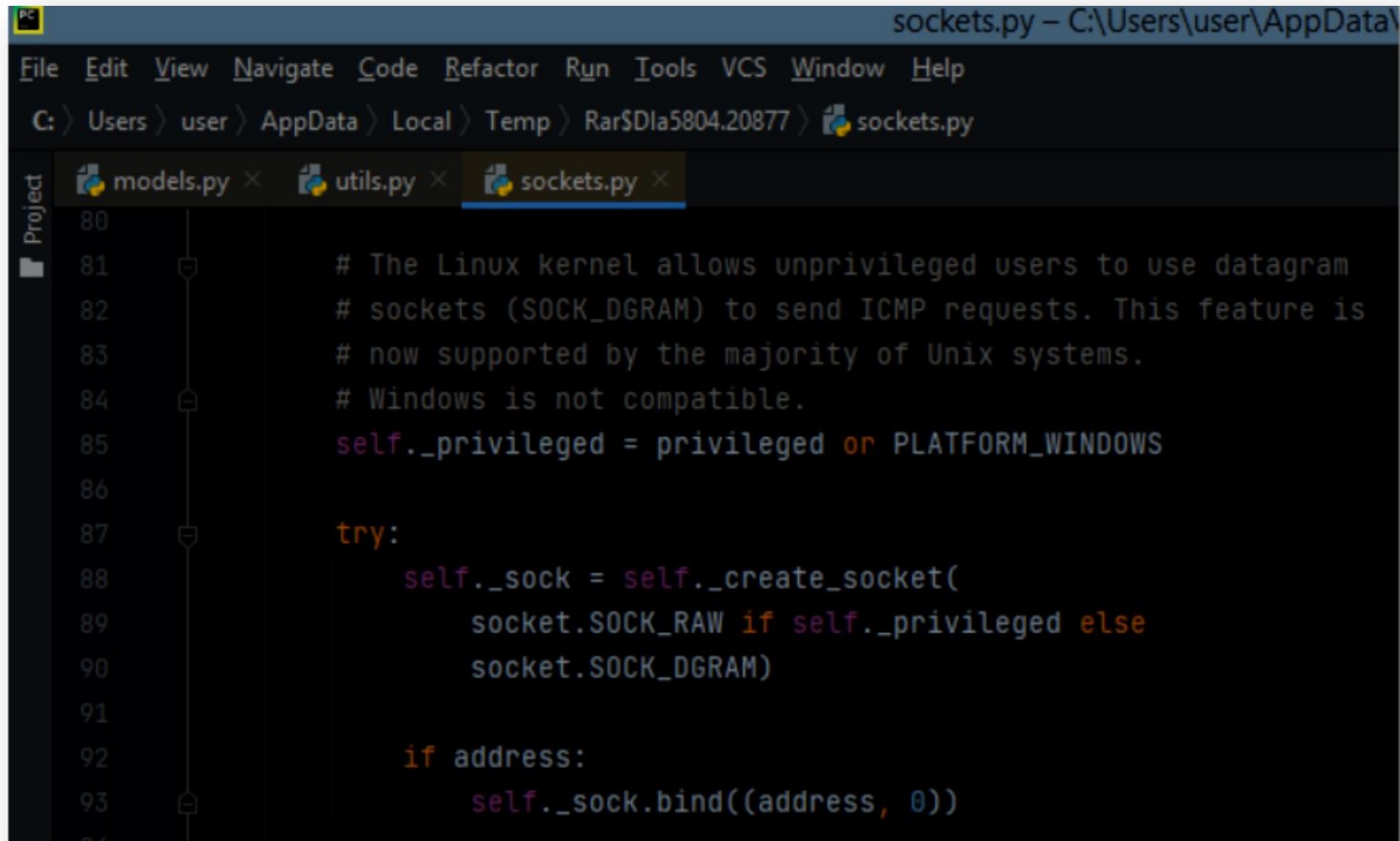
- **Import exceptions**

from icmplib import ICMPLibError, NameLookupError, ICMPSocketError

from icmplib import SocketAddressError, SocketPermissionError

from icmplib import SocketUnavailableError, SocketBroadcastError, TimeoutExceeded

from icmplib import ICMPError, DestinationUnreachable, TimeExceeded



```
sockets.py - C:\Users\user\AppData\
File Edit View Navigate Code Refactor Run Tools VCS Window Help
C:\Users\user\AppData\Local\Temp\Rar$DIa5804.20877\sockets.py
models.py x utils.py x sockets.py x
Project
80
81 # The Linux kernel allows unprivileged users to use datagram
82 # sockets (SOCK_DGRAM) to send ICMP requests. This feature is
83 # now supported by the majority of Unix systems.
84 # Windows is not compatible.
85 self._privileged = privileged or PLATFORM_WINDOWS
86
87 try:
88     self._sock = self._create_socket(
89         socket.SOCK_RAW if self._privileged else
90         socket.SOCK_DGRAM)
91
92     if address:
93         self._sock.bind((address, 0))
94
```

1. Ping
2. Broadcast Ping
3. Multi-Ping
4. Traceroute
5. Verbose Ping
6. Verbose traceroute