



Network Programming for Windows 02:

# Winsock Design

jintaeks@dongseo.ac.kr  
Division of Digital Contents, DSU

# Winsock Design

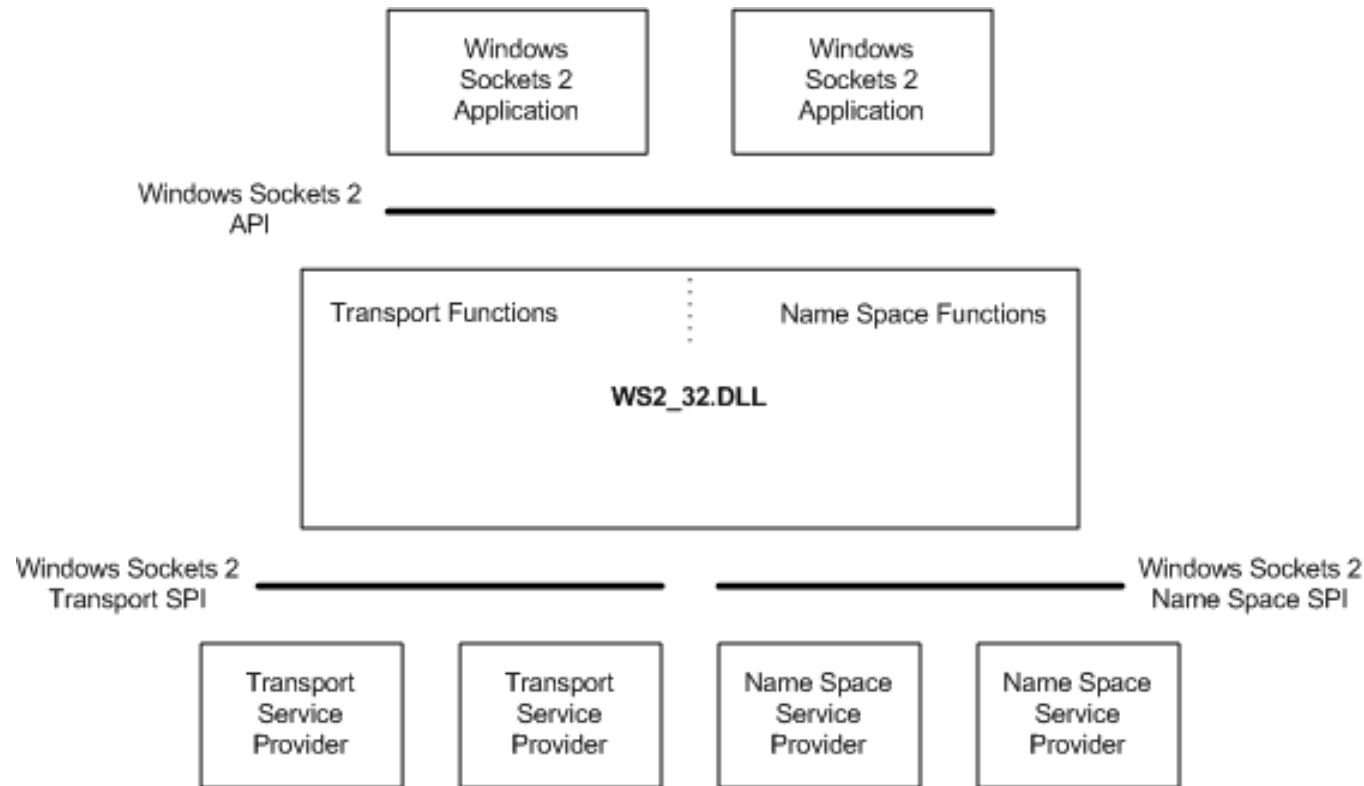
- ✓ System Architecture
- ✓ Protocol Characteristics
- ✓ Winsock Catalog
- ✓ References

- 
- ✓ We'll discuss protocol characteristics and how applications can enumerate the installed protocols.
  - ✓ Then we'll discuss the details of socket creation via the **socket** and **WSASocket** functions and how they interact with the **Winsock catalog**.

# System Architectures

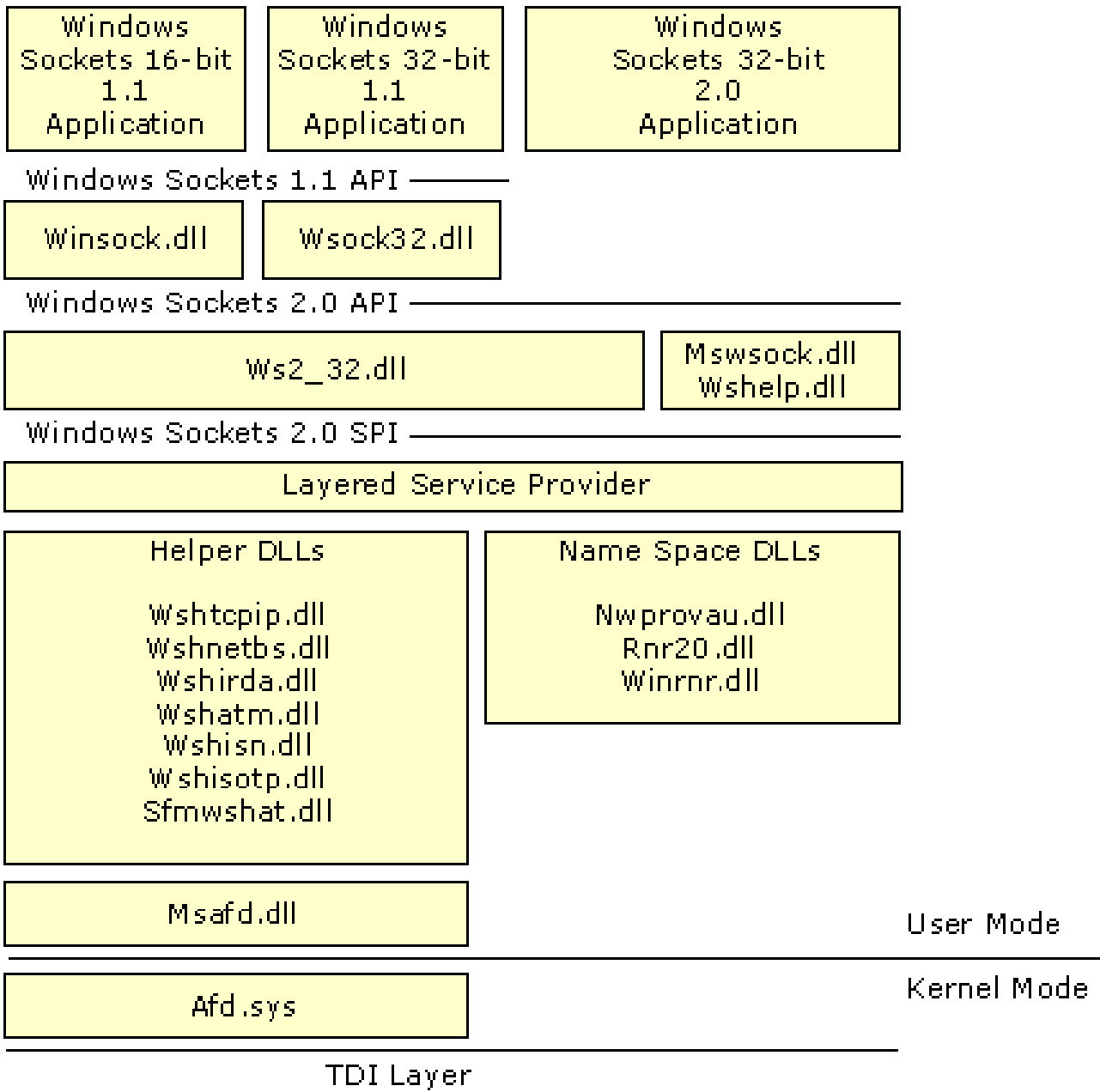
- ✓ The majority of the Winsock API is implemented in **WS2\_32.DLL** and is declared in **WINSOCK2.H**.
- ✓ The only exception is for the Microsoft-specific Winsock extensions (such as TransmitFile, AcceptEx, etc.), which are located in **MSWSOCK.DLL**.

- ✓ The Windows Sockets 2 architecture is compliant with the Windows Open System Architecture (WOSA).



- ✓ Winsock defines a standard **service provider interface (SPI)** between the application programming interface (API), with its functions exported from WS2\_32.dll and the protocol stacks.

# Required DLLs

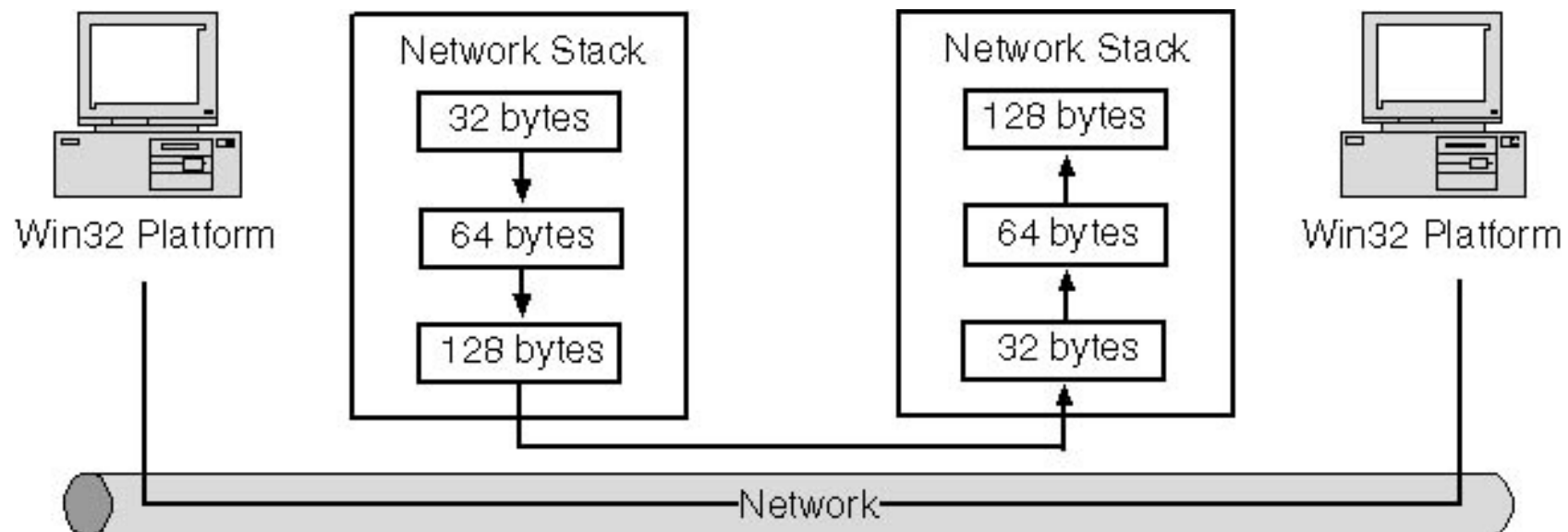


# Protocol Characteristics

- ✓ A multitude of different transport protocols are available on Windows, such as TCP, UDP, IPX, and SPX.
- ✓ Some require a connection to be established before sending or receiving data. Others don't guarantee the **reliability** or **integrity** of the data.

# Message-Oriented

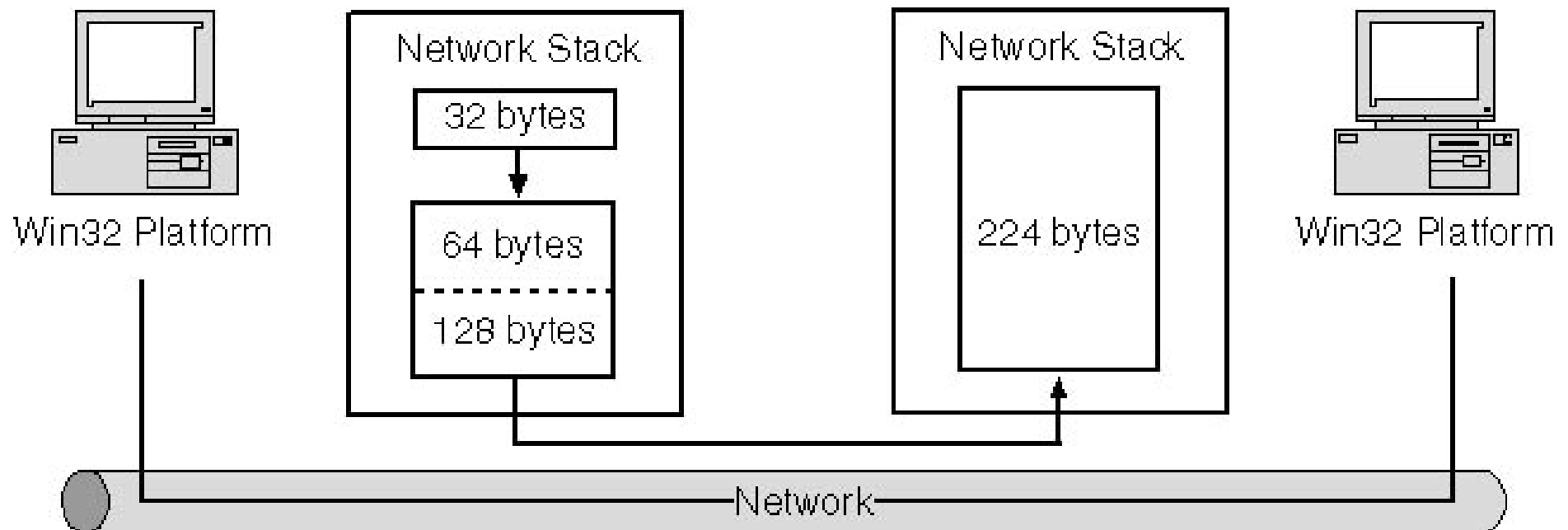
- ✓ For each discrete write command, **it transmits only those bytes as a single message** on the network.
- ✓ When the receiver requests data, the data returned is a discrete message written by the sender.





# Stream-Oriented

- ✓ A protocol that does not preserve message boundaries is often referred to as a **stream-based** protocol.



# Nagle Algorithm

- ✓ **Nagle's algorithm** is a means of improving the efficiency of [TCP/IP](#) networks by reducing the number of packets that need to be sent over the network.
- ✓ It was defined by John Nagle.
- ✓ "small-packet problem": where an application repeatedly emits data in small chunks, frequently only 1 [byte](#) in size.
- ✓ Since [TCP](#) packets have a 40-byte header (20 bytes for TCP, 20 bytes for [IPv4](#)), this results in a 41-byte packet for 1 byte of useful information, a huge overhead.

---

```
if there is new data to send
  if the window size  $\geq$  MSS and available data is  $\geq$  MSS
    send complete MSS segment now
  else
    if there is unconfirmed data still in the pipe
      enqueue data in the buffer until an acknowledge is received
    else
      send data immediately
    end if
  end if
end if
```

# Pseudo Stream

---

- ✓ Pseudo stream is a term often applied to a system with a message-based protocol that sends data in discrete packets, which the receiver reads and buffers in a pool so the receiving application reads data chunks of any size.

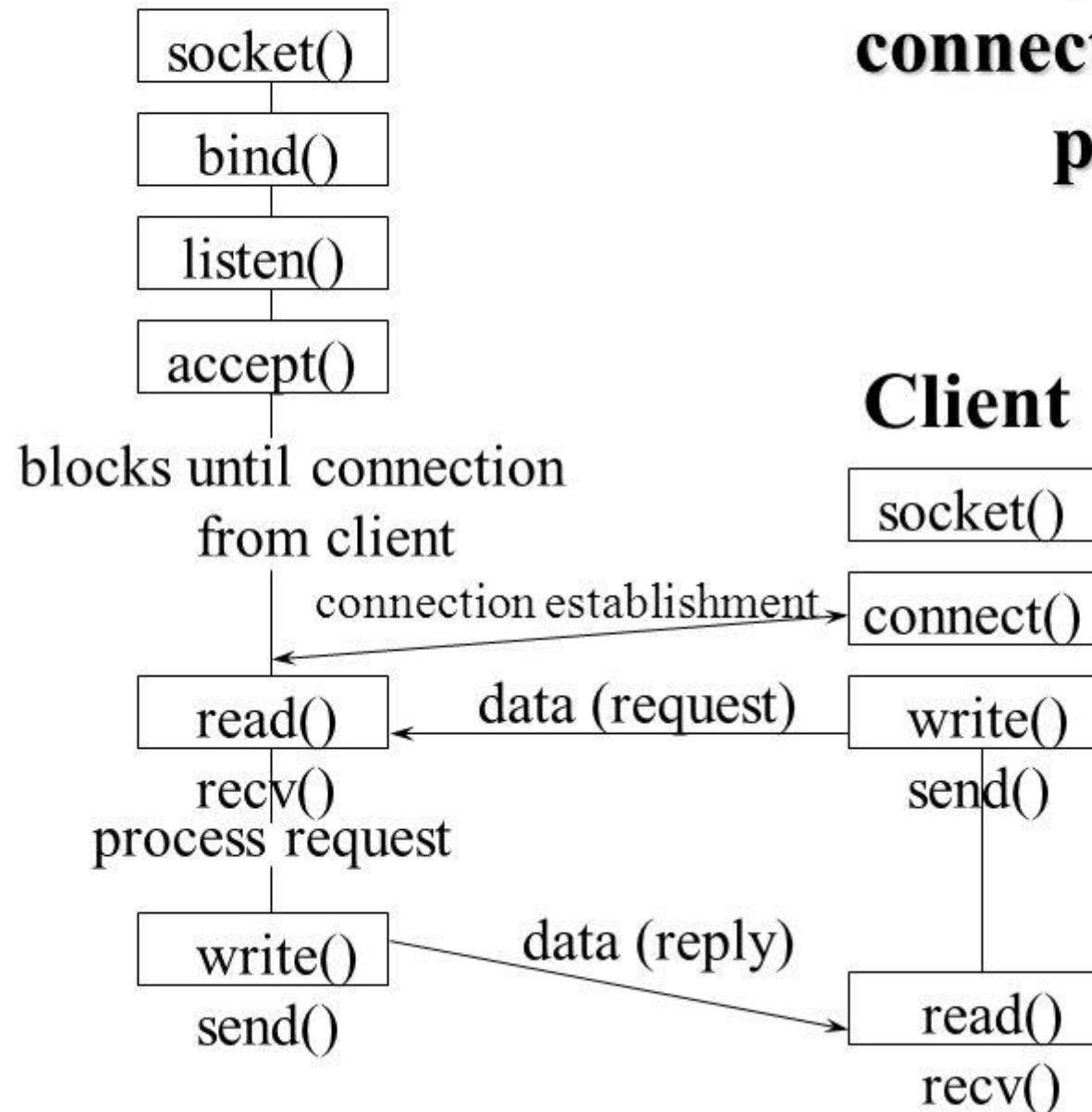
# Connection-Oriented and Connectionless

---

- ✓ In **connection-oriented** services, a **path** is established between the two communicating parties before any data is exchanged.
- ✓ This ensures that there is a **route** between the two parties in addition to ensuring that both parties are alive and responding.
- ✓ On the other hand, a **connectionless** protocol makes no guarantees that the recipient is listening.

# Server

(connection-oriented protocol)

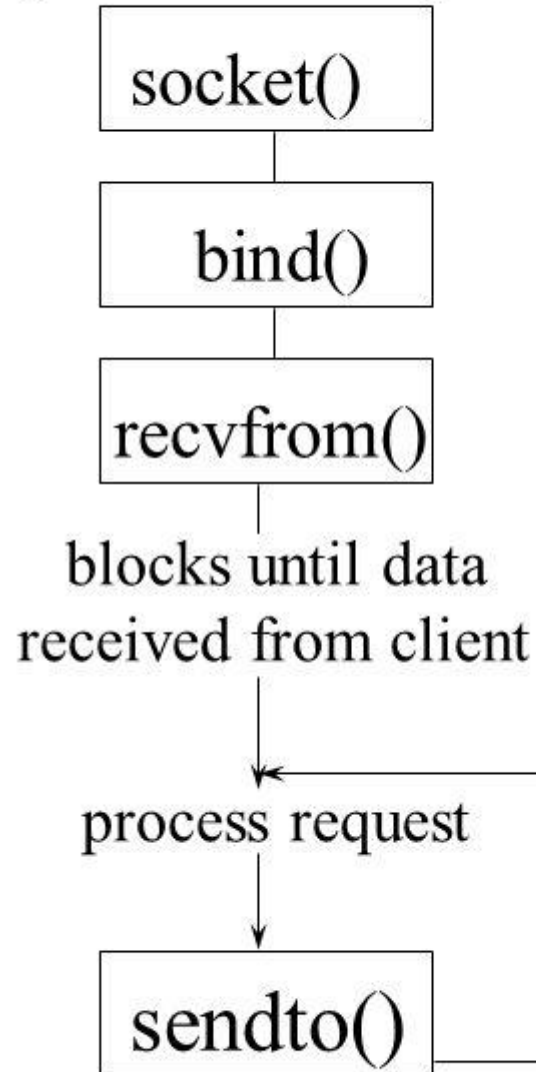


# Socket system calls for connection-oriented protocol

## Client

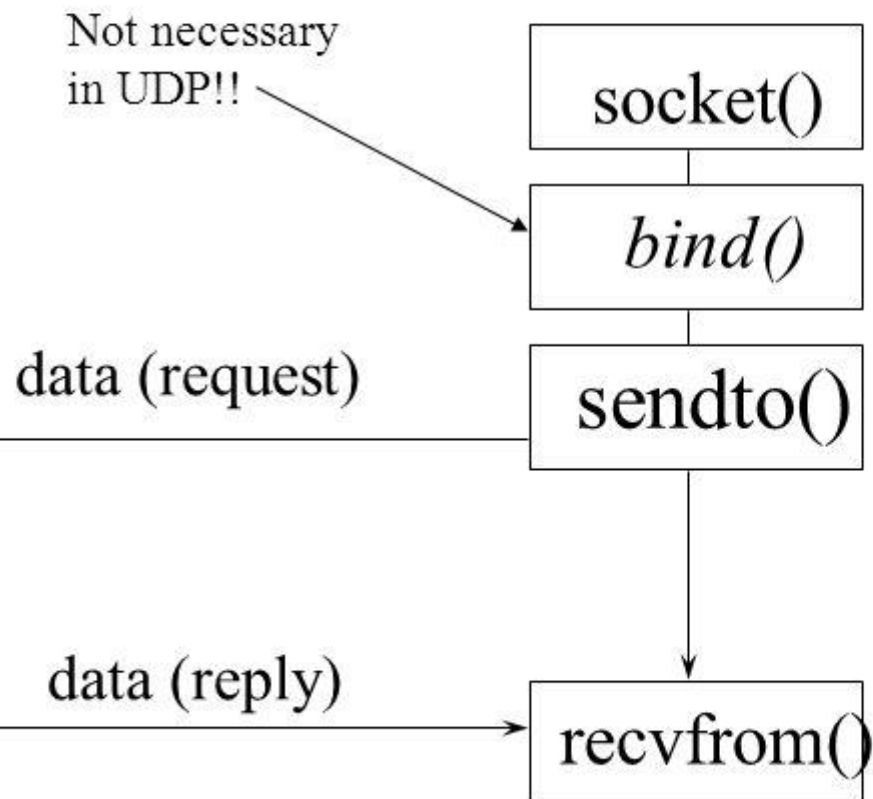
## Server

(connectionless protocol)



## Socket system calls for connectionless protocol

## Client



# Reliability and Ordering

- ✓ **Reliability**, or guaranteed delivery, ensures that each byte of data from the sender will reach the intended recipient unaltered.
- ✓ **Ordering** has to do with the order in which the data arrives at the recipient. A protocol that preserves ordering ensures that the recipient receives the data in the exact order that it was sent.
- ✓ In most cases, connection-oriented protocols do guarantee reliability.
- ✓ TCP doesn't guarantee the order.



# Graceful Close

---

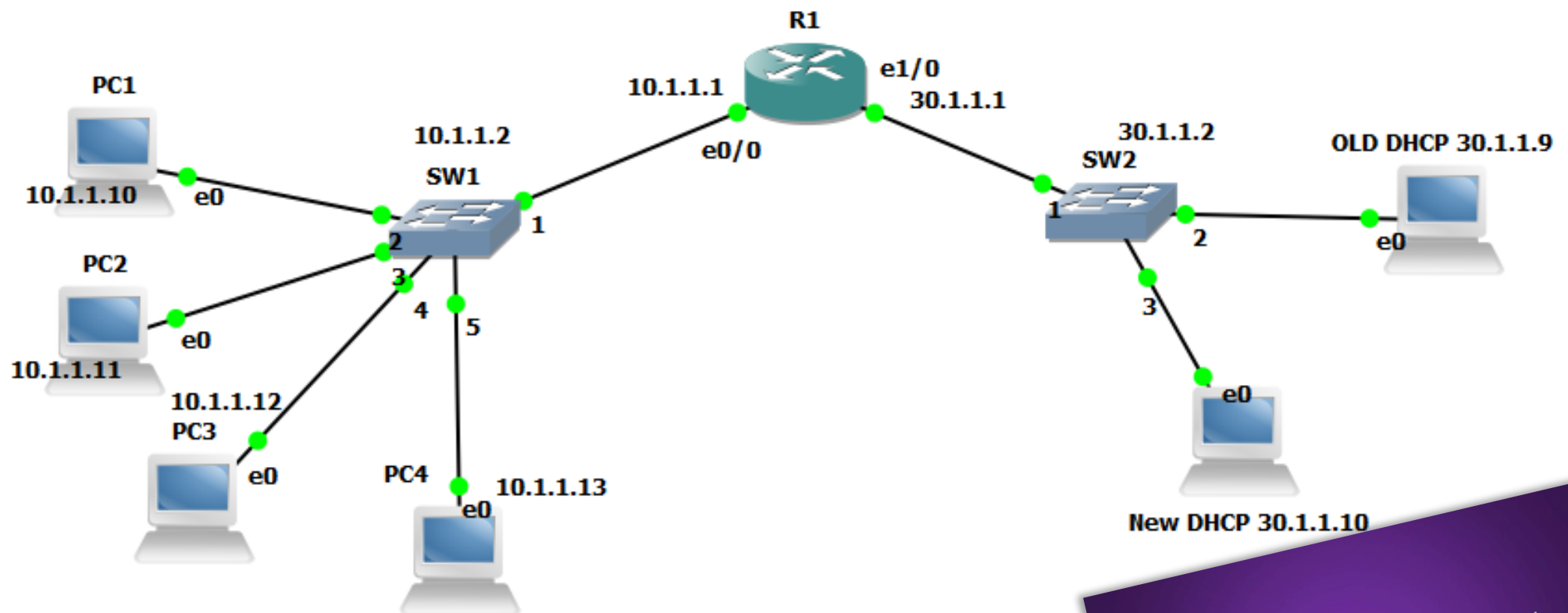
- ✓ A graceful close is associated with **connection-oriented protocols only**.
- ✓ In a graceful close, one side initiates the shutting down of a communication session and the other side still has the opportunity to read pending data on the wire or the network stack.

# Broadcast Data

---

- ✓ To broadcast data is to send data from one workstation so that all other workstations on the LAN can receive it.
- ✓ This feature is available to connectionless protocols because all machines on the LAN can pick up and process a broadcast message.

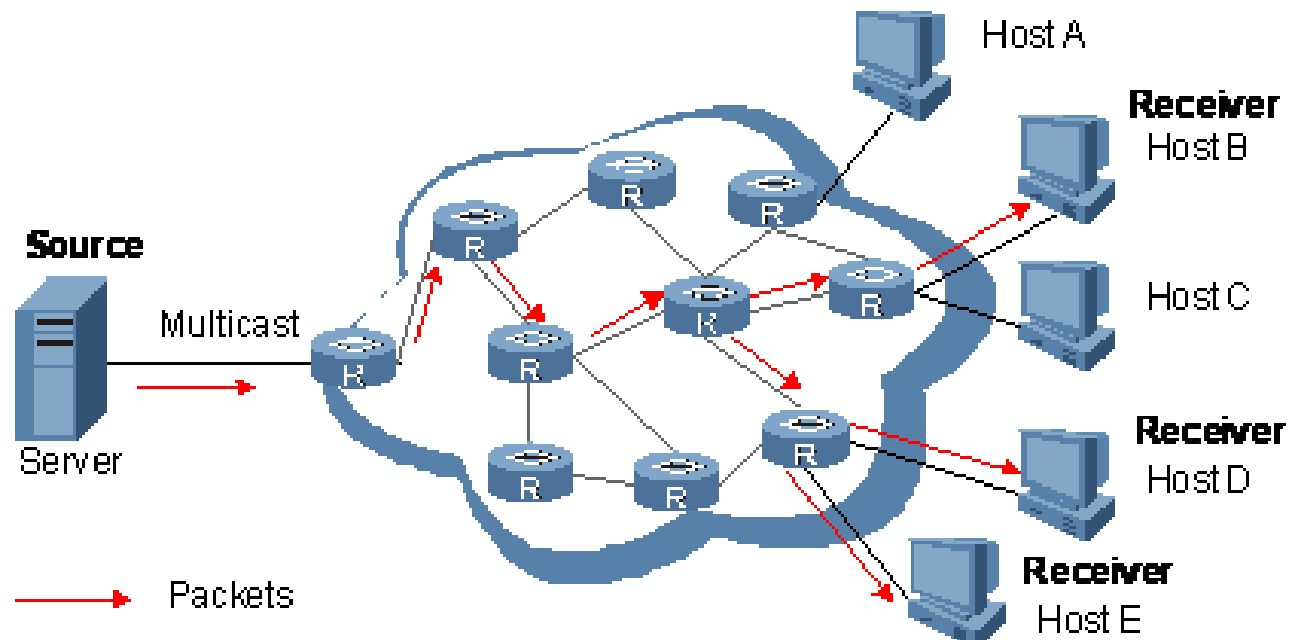
# Ex) Broadcast of DHCP Protocol



Dynamic Host Configuration Protocol  
How a Computer Gets Its IP Address

# Multicast Data

- ✓ Multicasting is the capability of one process to send data that one or more recipients will receive.



# Quality of Service (QOS)

---

- ✓ QOS is an application's capability to request certain network bandwidth requirements to be dedicated for exclusive use.
- ✓ One good use for QOS is real-time video streaming.

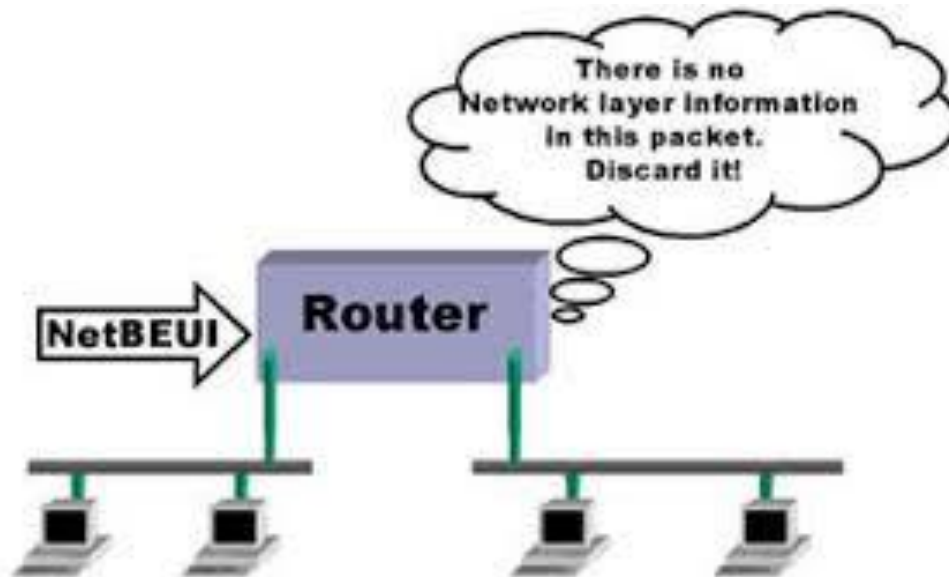
# Partial Message

---

- ✓ Partial messages apply to **message-oriented protocols only**.
- ✓ Let's say an application wants to receive a message but the local computer has received only part of the data.

# Routing Considerations

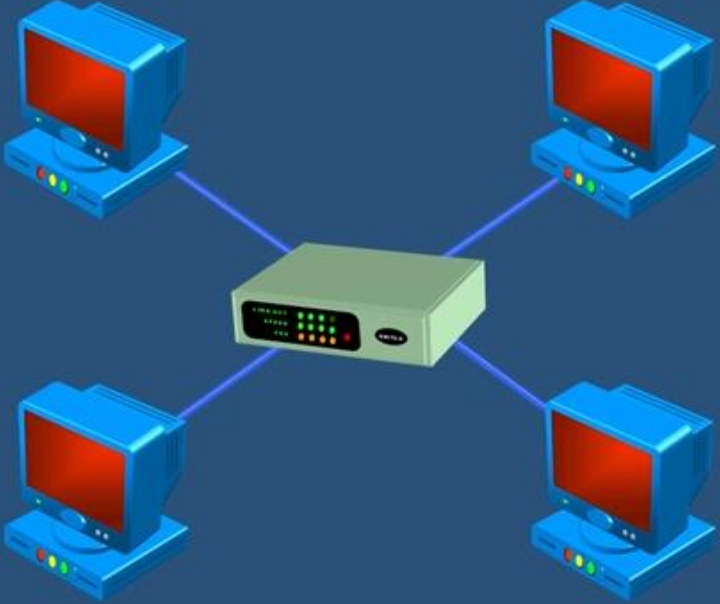
- ✓ If a protocol is **routable**, a successful communication path can be set up (either a **virtual** connection-oriented **circuit** or a **data path** for datagram communication) between two workstations, no matter what network hardware lies between them.
- ✓ **NetBEUI** is the only protocol supported by Windows platforms that is not capable of being routed.



# Difference between Hub, Switch and Router

POWERCERT VIDEOS

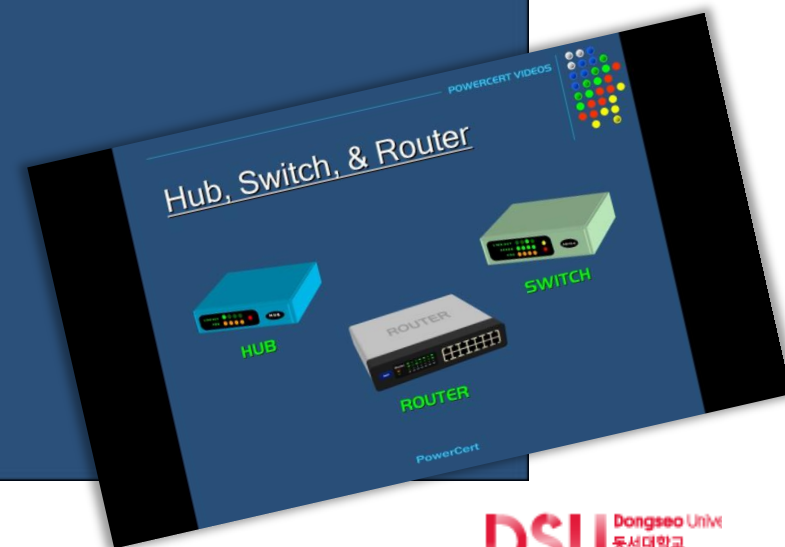
## Hub, Switch, & Router



Hubs and Switches are used to exchange data within a local area network.

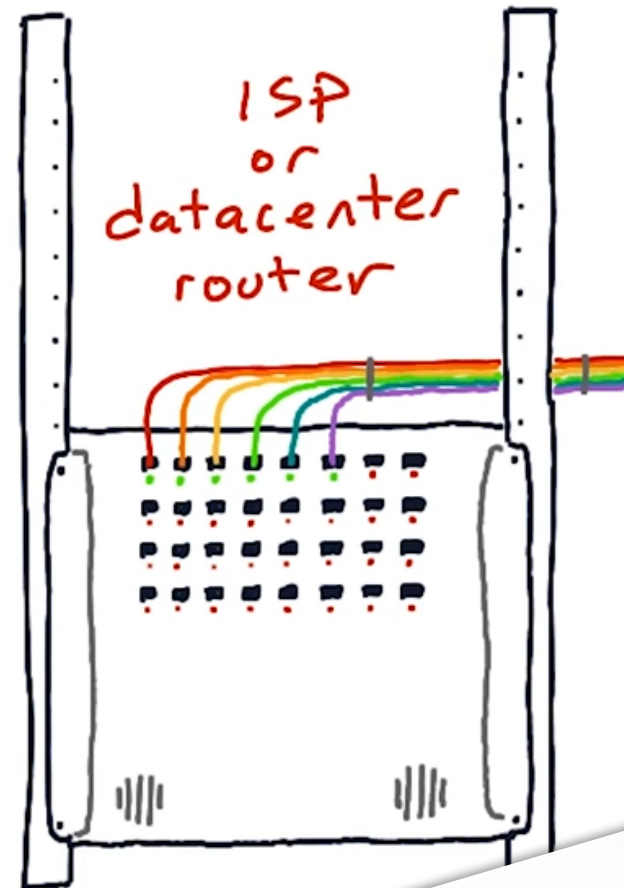
PowerCert

The diagram illustrates a central network device (Hub or Switch) connected to four desktop computers. The device is a light green rectangular box with multiple ports and indicator lights. Four blue lines represent network cables connecting the central device to each of the four desktop computers, which are shown as blue monitors on stands with red screens. The background is a dark blue gradient. In the top right corner, there is a logo consisting of a grid of colored dots (white, blue, green, red, yellow) arranged in a pattern. The text 'POWERCERT VIDEOS' is in the top right. The title 'Hub, Switch, & Router' is in the center, underlined. The text 'Hubs and Switches are used to exchange data within a local area network.' is to the right of the diagram. The 'PowerCert' logo is at the bottom center.

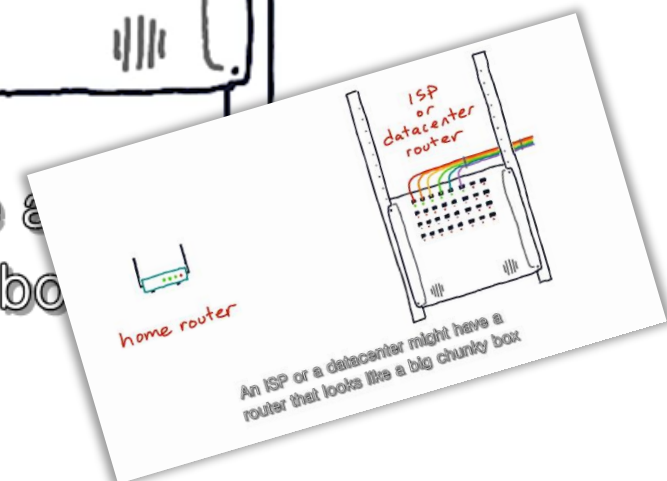




# Routers and Default Gateway



An ISP or a datacenter might have a router that looks like a big chunky box



# Winsock Catalog

- ✓ The Winsock catalog is a **database that contains the different protocols available on the system.**
- ✓ Winsock provides a method for determining which protocols are installed on a given workstation and returning a variety of characteristics for each protocol.

```
int WSAEnumProtocols(  
    LPINT lpiProtocols,  
    LPWSA_PROTOCOL_INFO lpProtocolBuffer,  
    LPDWORD lpdwBufferLength  
);
```

```

typedef struct _WSAPROTOCOL_INFO {
    DWORD          dwServiceFlags1;
    DWORD          dwServiceFlags2;
    DWORD          dwServiceFlags3;
    DWORD          dwServiceFlags4;
    DWORD          dwProviderFlags;
    GUID           ProviderId;
    DWORD          dwCatalogEntryId;
    WSAPROTOCOLCHAIN ProtocolChain;
    int            iVersion;
    int            iAddressFamily;
    int            iMaxSockAddr;
    int            iMinSockAddr;
    int            iSocketType;
    int            iProtocol;
    int            iProtocolMaxOffset;
    int            iNetworkByteOrder;
    int            iSecurityScheme;
    DWORD          dwMessageSize;
    DWORD          dwProviderReserved;
    TCHAR          szProtocol[ WSAPROTOCOL_LEN + 1 ];
} WSAPROTOCOL_INFO, FAR * LPWSAPROTOCOL_INFO;

```

# Practice

- ✓ chapter02→enumcat project
- ✓ SimpleWinsockClientServer project

```
C:\Users\13FGames>tracert www.google.com
```

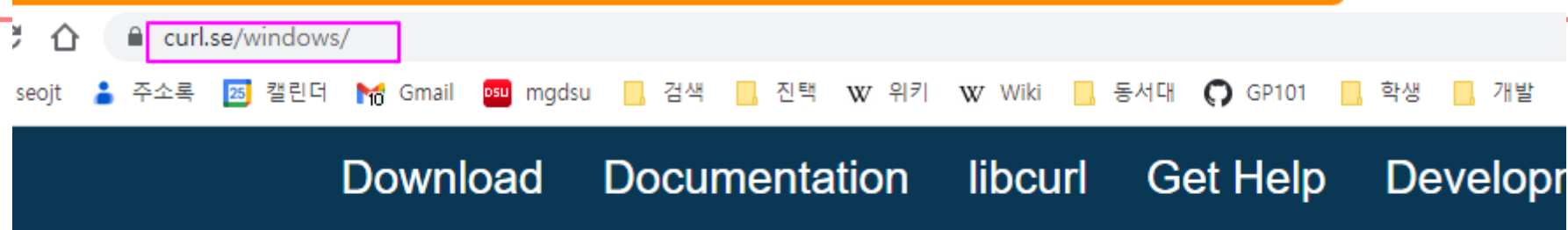
```
Tracing route to www.google.com [172.217.24.132]  
over a maximum of 30 hops:
```

1	1 ms	1 ms	3 ms	192.168.0.1
2	81 ms	3 ms	3 ms	222.96.24.1
3	3 ms	3 ms	3 ms	112.191.9.233
4	*	*	*	Request timed out.
5	*	*	*	Request timed out.
6	9 ms	11 ms	9 ms	112.174.5.118
7	45 ms	45 ms	49 ms	72.14.194.194
8	38 ms	40 ms	38 ms	108.170.242.161
9	41 ms	40 ms	40 ms	72.14.234.229
10	40 ms	40 ms	40 ms	nrt20s01-in-f4.1e100.net [172.217.24.132]

```
Trace complete.
```

```
C:\Users\13FGames>
```

# Download and Install Curl.exe




[curl](#) / [Download](#) / **Windows downloads**

## curl 7.86.0 for Windows


These are the latest and most up to date **official** curl binary builds for Microsoft Windo

**curl version:** 7.86.0  
**Build:** 7.86.0\_2  
**Date:** 2022-10-26  
**Changes:** [7.86.0 changelog](#)



### curl for 64-bit

Size: 10.1 MB  
sha256: 1175599e2c8a26fdfa981064367bfe8e07cb17457eb6027f640f5a6d



### curl for 64-bit (ARM)

Size: 8.7 MB

```
C:\Users\13FGames>tracert www.google.com
Tracing route to www.google.com [172.217.24.132]
over a maximum of 30 hops:
```

```
  1      1 ms      1 ms      3 ms  192.168.0.1
  2     81 ms     3 ms     3 ms  222.96.24.1
  3      3 ms     3 ms     3 ms  112.191.9.233
  4      *        *        *      Request timed out.
  5      *        *        *      Request timed out.
  6      9 ms     11 ms     9 ms  112.191.9.233
  7     45 ms     45 ms     49 ms  112.191.9.233
  8     38 ms     40 ms     38 ms  112.191.9.233
  9     41 ms     40 ms     40 ms  72.14.194.194
 10     40 ms     40 ms     40 ms  nrt20s01.inet10.google.com
```

Trace complete.

```
C:\Users\13FGames>curl ipinfo.io/222.96.24.1
```

```
{
  "ip": "222.96.24.1",
  "city": "Busan",
  "region": "Busan",
  "country": "KR",
  "loc": "35.1017,129.0300",
  "org": "AS4766 Korea Telecom",
  "postal": "48926",
  "timezone": "Asia/Seoul",
  "readme": "https://ipinfo.io/missingauth"
}
```

curl ipinfo.io/222.96.24.1

```
C:\Users\13FGames>curl ipinfo.io/112.174.5.118
```

```
{
  "ip": "112.174.5.118",
  "city": "Seoul",
  "region": "Seoul",
  "country": "KR",
  "loc": "37.5663,126.9784",
  "org": "AS129 Korea Telecom",
  "postal": "03186",
  "timezone": "Asia/Seoul",
  "readme": "https://ipinfo.io/missingauth"
}
```

```
C:\Users\13FGames>curl ipinfo.io/72.14.194.194
```

```
{
  "ip": "72.14.194.194",
  "city": "Mountain View",
  "region": "California",
  "country": "US",
  "loc": "37.4056,-122.0775",
  "org": "AS15169 Google LLC",
  "postal": "94043",
  "timezone": "America/Los_Angeles",
  "readme": "https://ipinfo.io/missingauth"
}
```

```
C:\Users\13FGames>
```

# Practice

- ✓ Install curl.exe
- ✓ Add command line path for curl.exe
- ✓ Find a network path to [www.facebook.com](http://www.facebook.com) with tracert.exe
  - tracert www.facebook.com
- ✓ Find the location of the first router on the path and spot the location on the google map.
  - curl ipinfo.io/[ip-address]



53°19'59.2"N 6°14'56.0"W

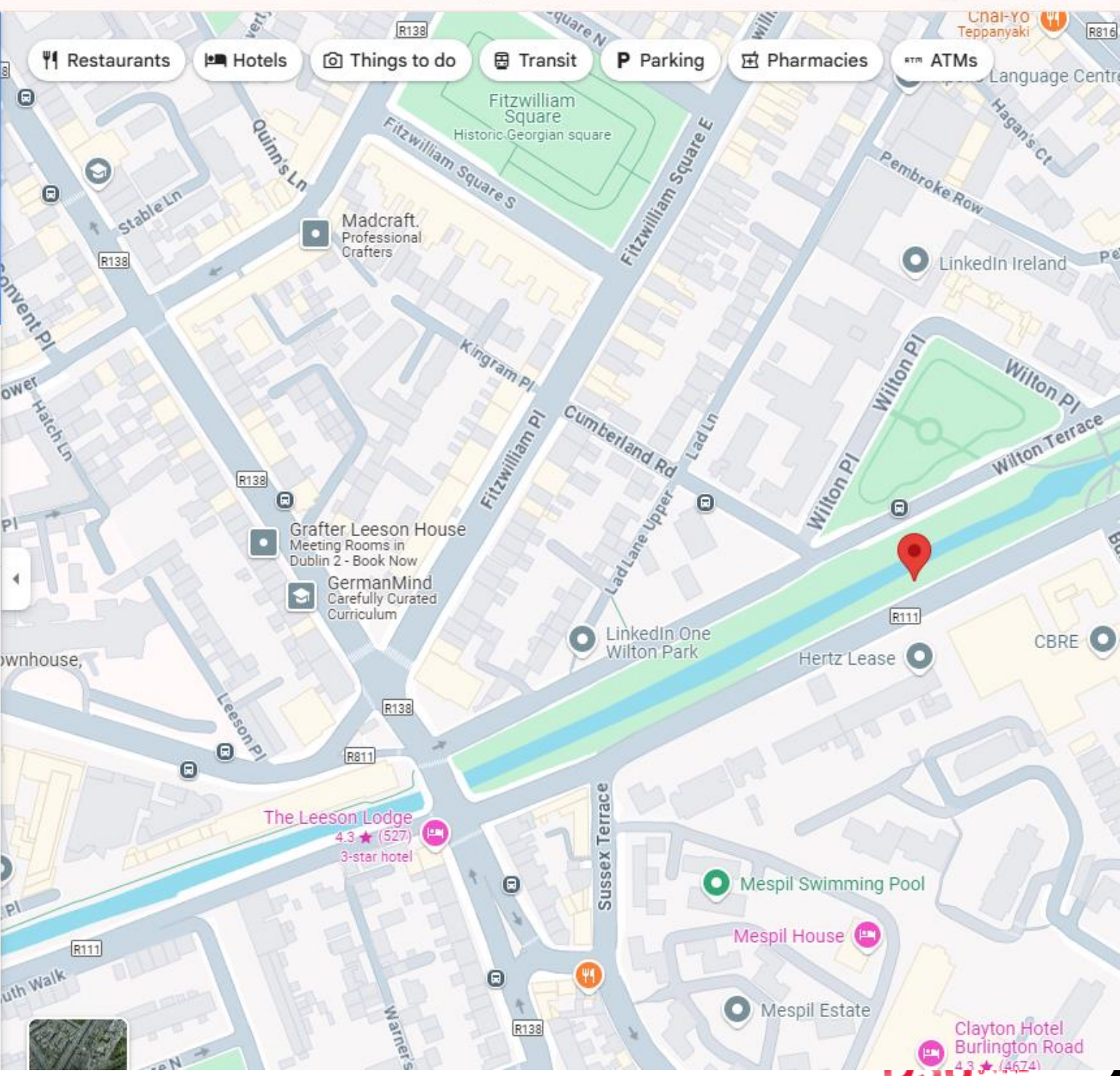


53°19'59.2"N 6°14'56.0"W

53.333100, -6.248900

- Directions
- Save
- Nearby
- Send to phone
- Share

- 8QM2+6CX Dublin, Ireland
- Add a missing place
- Add your business
- Add a label
- Your Maps activity



# References

- ✓ <http://www.winsocketdotnetworkprogramming.com/winsock2programming/>

---

# MY **BRIGHT** FUTURE

**DSU** Dongseo University  
동서대학교