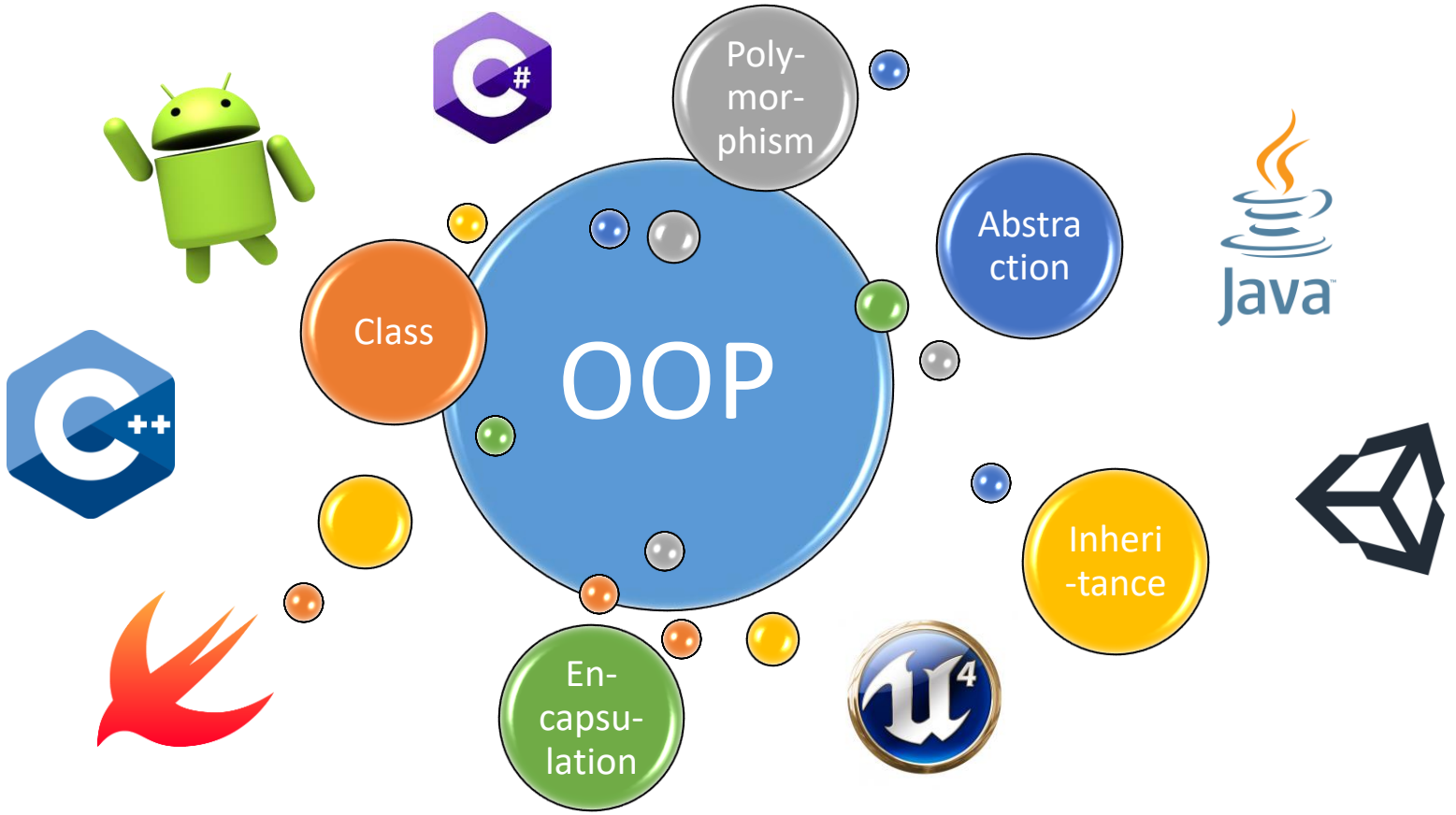



# Java Network Programming

Kuan-Ting Lai

2020/5/25

# Open Systems Interconnection (OSI) Model

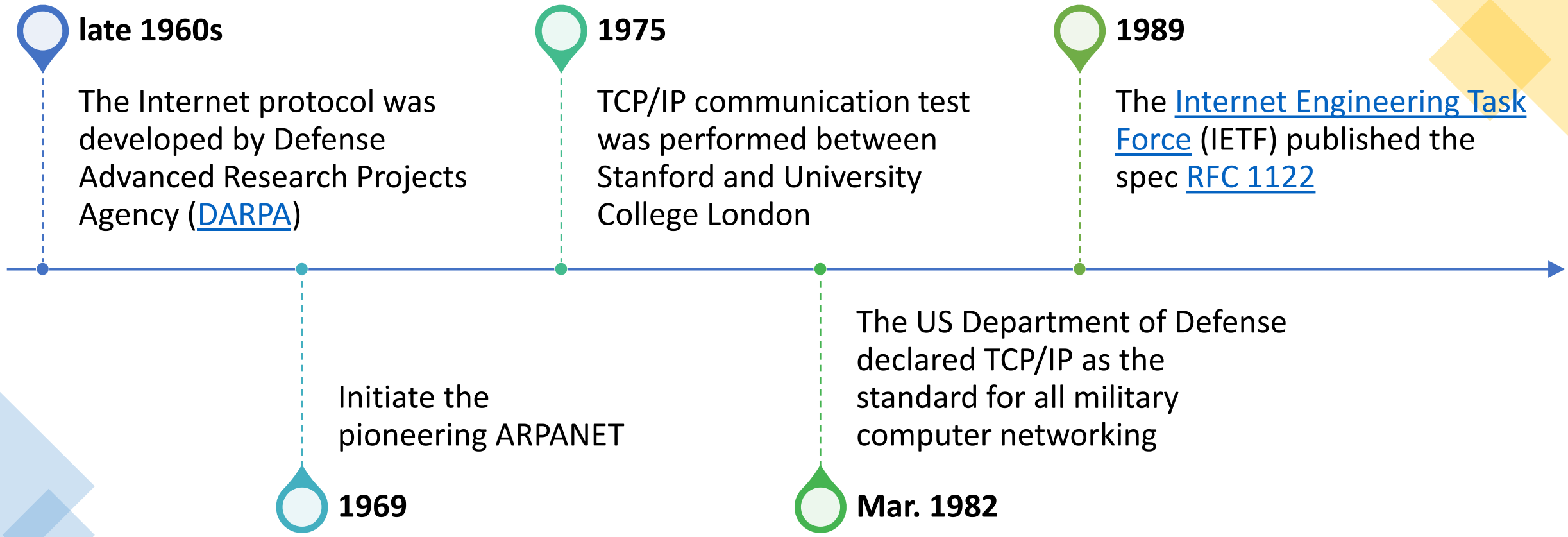
- Published in 1984 by the International Organization for Standardization (ISO), as standard ISO 7498
- Try to get companies to agree on common network standards to provide interconnections
- Define a reference model (OSI 7-layer model) and a set of protocols
- Predated by TCP/IP in the late 1980s

OSI (Open Source Interconnection) 7 Layer Model

| Layer | Application/Example | | Central Device/ Protocols | | DOD4 Model |
|---|---|---|---|---|---|
| **Application (7)**<br>Serves as the window for users and application processes to access the network services. | **End User layer** Program that opens what was sent or creates what is to be sent<br>Resource sharing • Remote file access • Remote printer access • Directory services • Network management | | **User Applications**<br>SMTP | GATEWAY<br>Can be used on all layers | Process |
| **Presentation (6)**<br>Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network. | **Syntax layer** encrypt & decrypt (if needed)<br>Character code translation • Data conversion • Data compression • Data encryption • **Character Set Translation** | | JPEG/ASCII<br>EBDIC/TIFF/GIF<br>PICT | | |
| **Session (5)**<br>Allows session establishment between processes running on different stations. | **Synch & send to ports** (logical ports)<br>Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc. | | **Logical Ports**<br>RPC/SQL/NFS<br>NetBIOS names | | |
| **Transport (4)**<br>Ensures that messages are delivered error-free, in sequence, and with no losses or duplications. | **TCP** Host to Host, Flow Control<br>Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing | PACKET FILTERING | TCP/SPX/UDP | | Host to Host |
| **Network (3)**<br>Controls the operations of the subnet, deciding which physical path the data takes. | **Packets** ("letter", contains IP address)<br>Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting | | **Routers**<br>IP/IPX/ICMP | | Internet |
| **Data Link (2)**<br>Provides error-free transfer of data frames from one node to another over the Physical layer. | **Frames** ("envelopes", contains MAC address)<br>[NIC card — Switch — NIC card] (end to end)<br>Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control | | **Switch Bridge WAP**<br>PPP/SLIP | Land Based Layers | Network |
| **Physical (1)**<br>Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium. | **Physical structure** Cables, hubs, etc.<br>Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts | | **Hub** | | |

https://sites.google.com/site/yutbms/osi-model

# Internet Protocol Suite (TCP/IP)

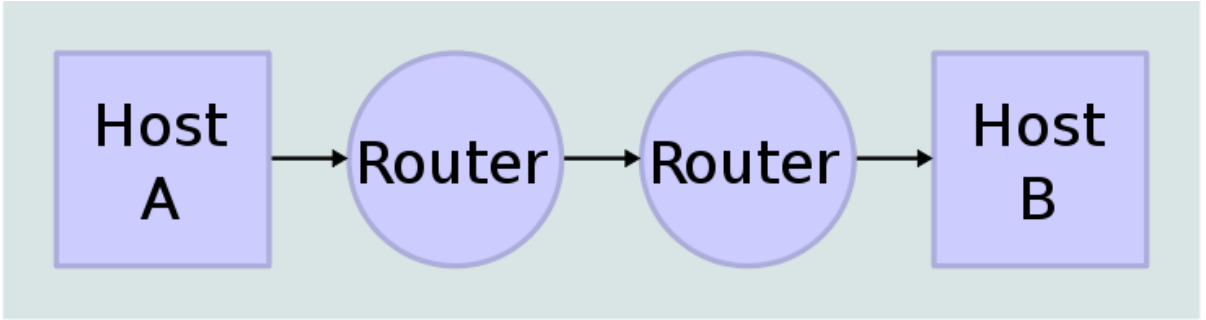


# OSI Model vs. TCP/IP Model

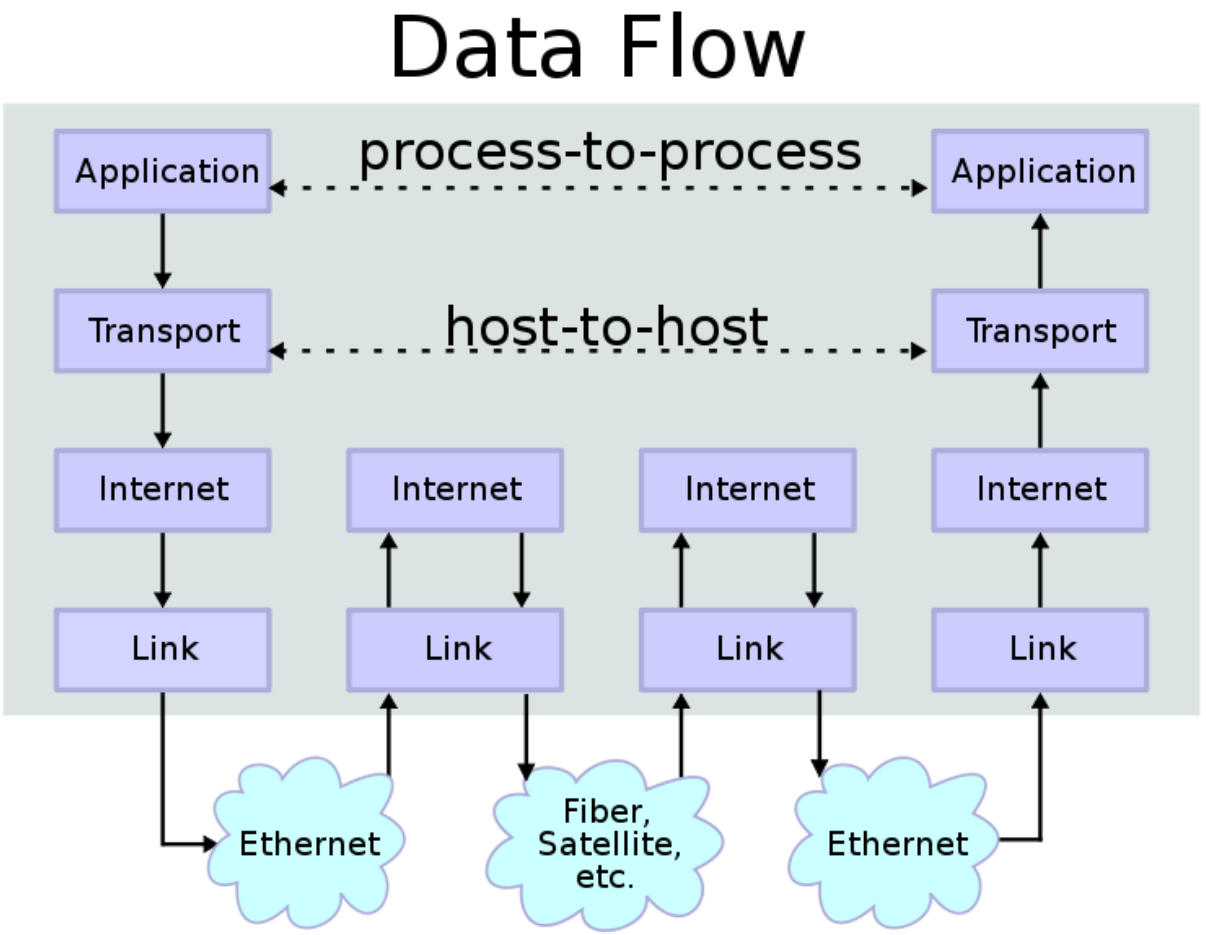
| OSI Model | TCP/IP Model |
|---|---|
| Application Layer | Application layer |
| Presentation Layer | |
| Session Layer | |
| Transport Layer | Transport Layer |
| Network Layer | Internet Layer |
| Data link layer | Link Layer |
| Physical layer | |

http://mhshohag.com/compare-and-contrast-osi-and-tcp-ip-models/

# Network Topology

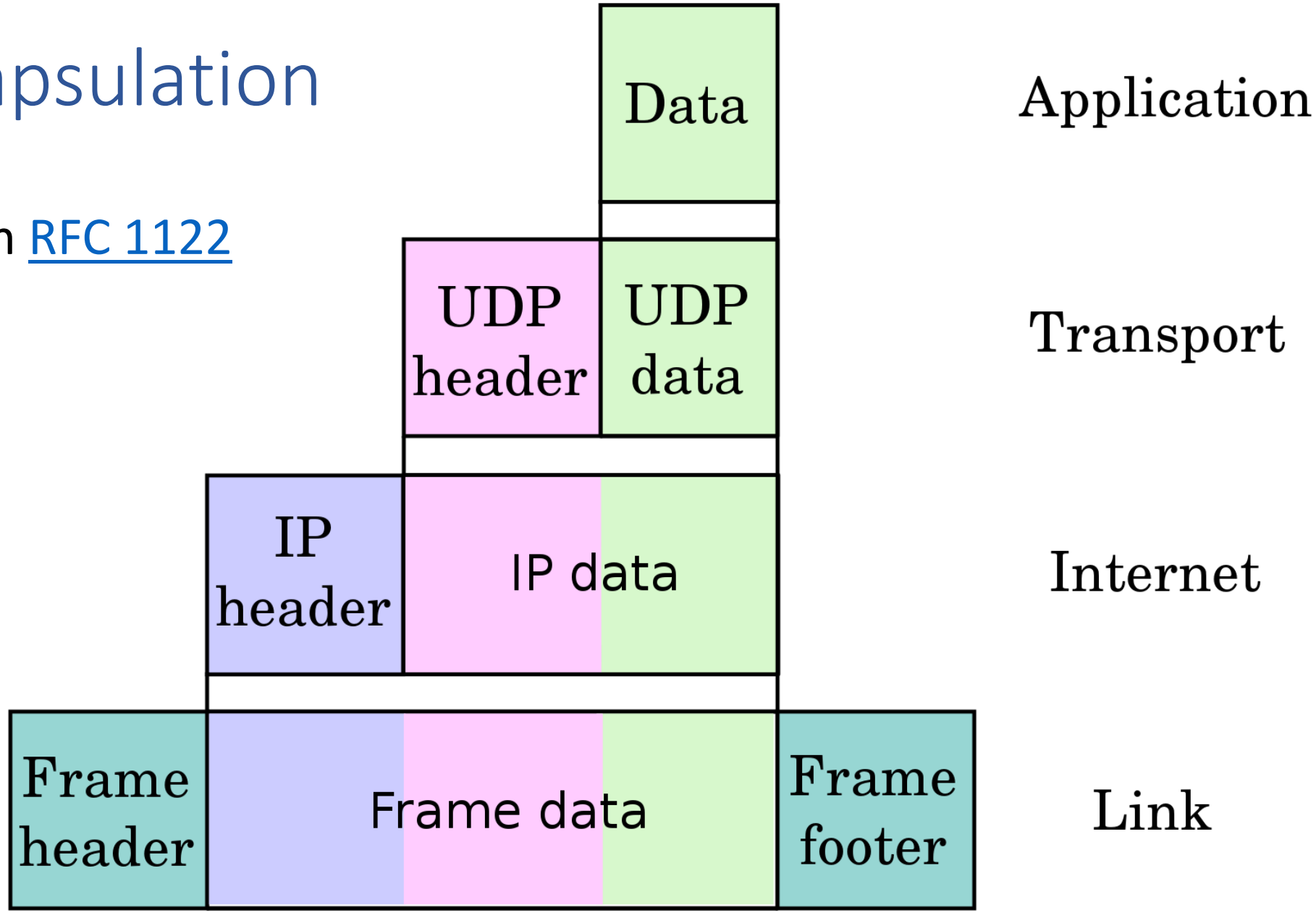



- Application layer
  - HTTP, FTP, SSH, SMTP
- Transport layer
  - Transmission Control Protocol (TCP)
  - User Datagram Protocol (UDP)
- Internet layer
  - Internet Protocol, ICMP, IGMP (multicast)
  - IP addressing (IPv4, IPv6)
- Link layer
  - Media Access Control (MAC) addresses
  - virtual private networks
  - networking tunnels



# Data Encapsulation

- Data Packet in RFC 1122



# TCP vs. UDP

| Transmission Control Protocol (**TCP**) | User Datagram Protocol (**UDP**) |
| --- | --- |
| Connection-oriented protocol | Connection-less protocol |
| Reliable. Data will be delivered | Unreliable. Data may be lost |
| Use flow control | No flow control |
| Data packets arrived in order | Data packets arrived in any order |
| Slower than UDP | Fast and low latency |

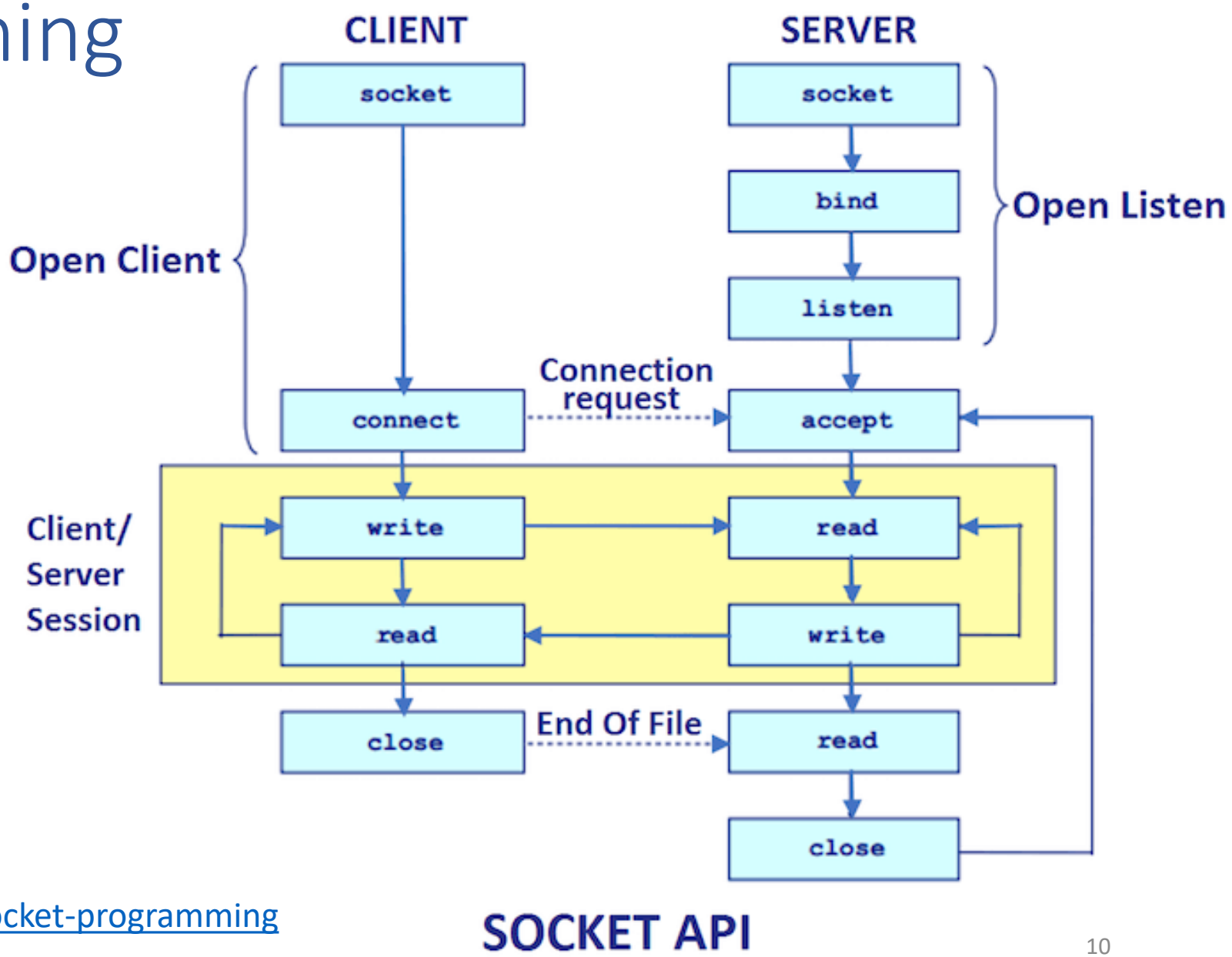
https://www.privateinternetaccess.com/blog/tcp-vs-udp-understanding-the-difference/

# TCP and UDP Applications

- TCP is best suited to be used for applications that require high reliability where timing is less of a concern.
  - World Wide Web (HTTP, HTTPS)
  - Secure Shell (SSH)
  - File Transfer Protocol (FTP)
  - Email (SMTP, IMAP/POP)
- UDP is best suited for applications that require speed and efficiency.
  - VPN tunneling
  - Streaming videos
  - Online games
  - Live broadcasts
  - Domain Name System (DNS)
  - Voice over Internet Protocol (VoIP)

# Socket Programming

- Client-server model
- Need to know IP address and port of server first




https://www.javatpoint.com/socket-programming

# Socket class

- Create  a Server socket and wait for a client connection

```
ServerSocket servSock = new ServerSocket(6666);
//establishes connection and waits for the client
Socket s = servSock.accept();
```

- Create a Client socket

```
Socket s = new Socket("localhost", 6666);
```

https://www.javatpoint.com/socket-programming

# EchoServer.java

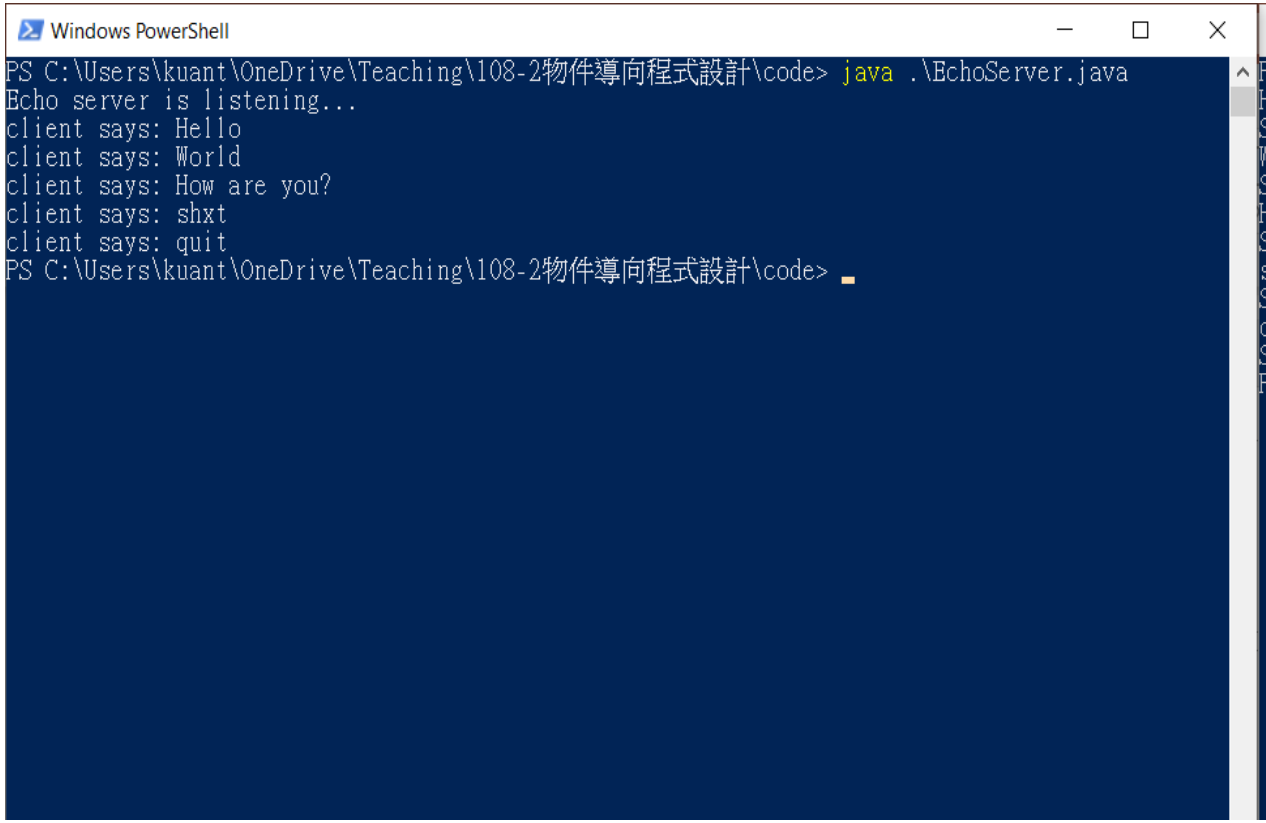
```
import java.net.*;
import java.io.*;
class EchoServer {
    public static void main(String args[]) throws Exception {
        ServerSocket servSock = new ServerSocket(6666);
        System.out.println("Echo server is listening...");
        Socket s = servSock.accept();
        DataInputStream din = new DataInputStream(s.getInputStream());
        DataOutputStream dout = new DataOutputStream(s.getOutputStream());

        String str="";
        while(!str.equals("quit")){
            str = din.readUTF();
            System.out.println("client says: " + str);
            dout.writeUTF(str);
            dout.flush();
        }
        din.close();
        s.close();
        servSock.close();
    }
}
```

# TCPClient.java

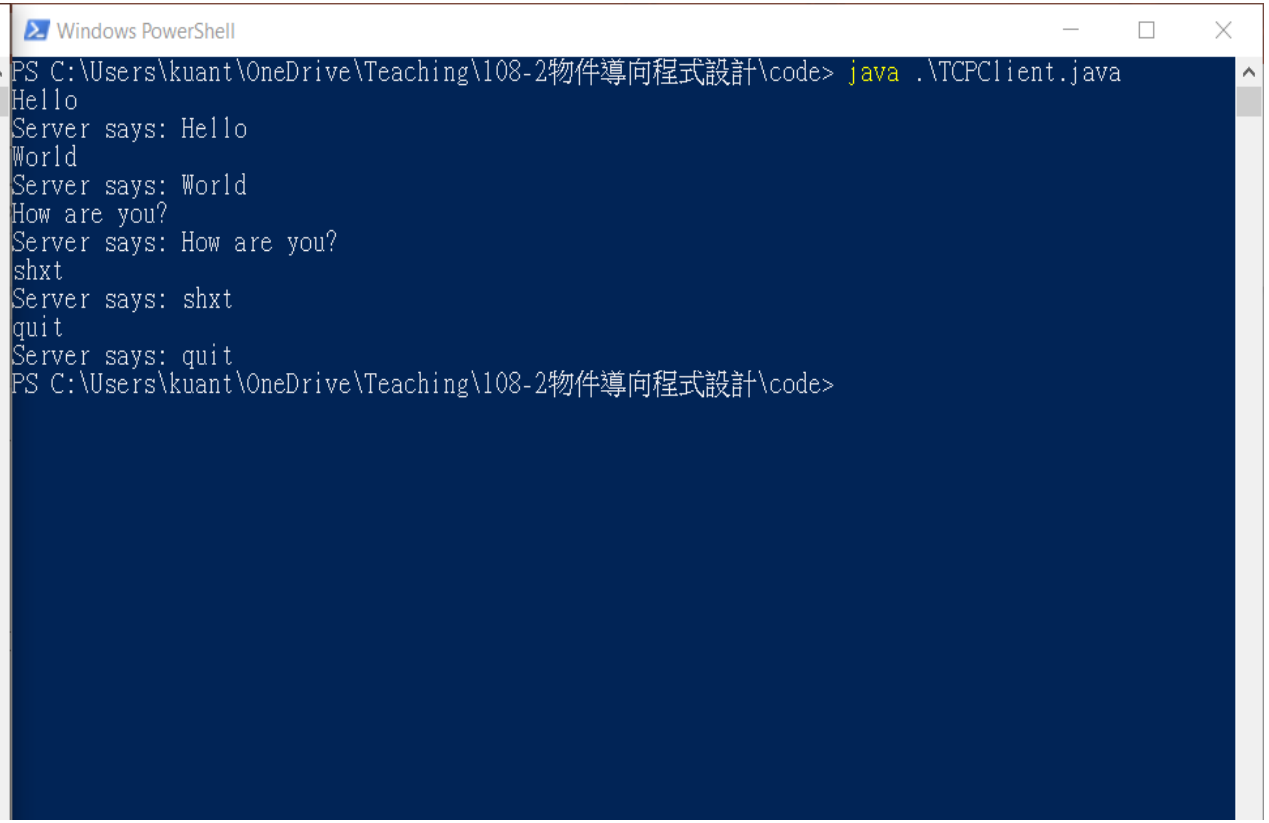
```
import java.net.*;
import java.io.*;

class TCPClient {
    public static void main(String args[]) throws Exception {
        Socket s = new Socket("localhost", 6666);
        DataInputStream din = new DataInputStream(s.getInputStream());
        DataOutputStream dout = new DataOutputStream(s.getOutputStream());
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        String str = "", str2 = "";
        while (!str.equals("quit")) {
            str = br.readLine();
            dout.writeUTF(str);
            dout.flush();
            str2 = din.readUTF();
            System.out.println("Server says: " + str2);
        }
        dout.close();
        s.close();
    }
}
```

# Test EchoServer and TCPClient

## EchoServer.java




## TCPClient.java



Windows PowerShell

```
PS C:\Users\kuant\OneDrive\Teaching\108-2物件導向程式設計\code> java .\EchoServer.java
```

Windows PowerShell

```
PS C:\Users\kuant\OneDrive\Teaching\108-2物件導向程式設計\code>
```

# ReadWeb.java

```
import java.net.*;
import java.io.*;
class ReadWeb {
    public static void main(String args[]) throws Exception {
        URL urlObject = new URL("http://www.aiotlab.org");
        URLConnection urlConnection = urlObject.openConnection();
        urlConnection.setRequestProperty("User-
                Agent", "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.11 (KHTML, l
                ike Gecko) Chrome/23.0.1271.95 Safari/537.11");
        System.out.println(toString(urlConnection.getInputStream()));
    }
    private static String toString(InputStream inputStream) throws IOException {
        try (BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(inpu
             tStream, "UTF-8")))
        {
            String inputLine;
            StringBuilder stringBuilder = new StringBuilder();
            while ((inputLine = bufferedReader.readLine()) != null)
            {
                stringBuilder.append(inputLine);
            }
            return stringBuilder.toString();
        }
    }}
```

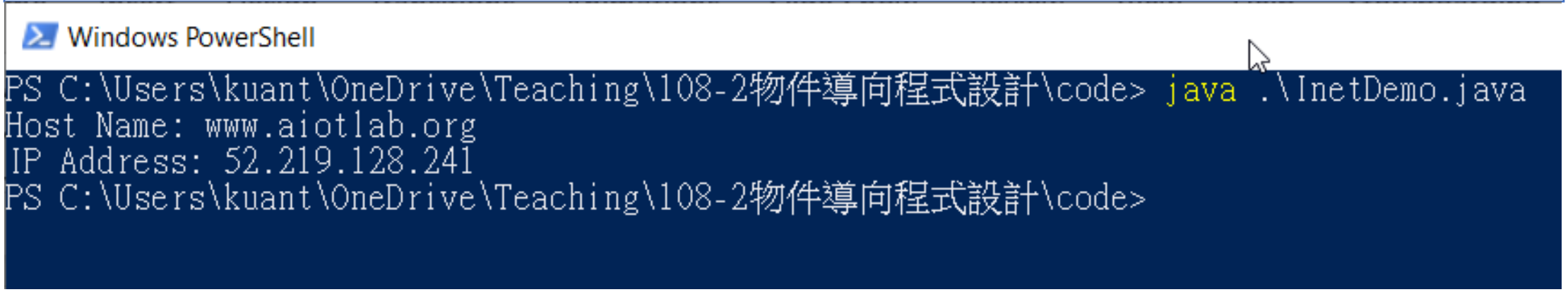
# Reading the HTML Source of a Web Page

Windows PowerShell

```
14</span></div>                </td>                <td class="gsc_a_c"><a                href="https://scholar.google.com.tw/scholar?oi=bibs&amp;hl=en&amp;cites=9993544465553274
5"                class="gsc_a_ac gs_ibl">94</a></td>                <td class="gsc_a_y"><span class="gsc_a_h gsc_a_hc gs_ibl">2014</span></td>                </tr>                <
tr class="gsc_a_tr">                <td class="gsc_a_t"><a                href="http://openaccess.thecvf.com/content_cvpr_2013/html/Liu_Sample-Specific_Late_Fusion_2013_CVPR_paper.h
tml"                data-href="/citations?view_op=view_citation&amp;hl=en&amp;user=5o3YVKQAAAAJ&amp;sortby=pubdate&amp;citation_for_view=5o3YVKQAAAAJ:Y5dfb0dijaUC"
  class="gsc_a_at">Sample-specific late fusion for visual category recognition</a>                <div class="gs_gray">D Liu, KT Lai, G Ye, MS Chen, SF Chang</div>                <d
iv class="gs_gray">Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on,                803-810<span class="gs_oph">, 2013</span></div>                </td>
                <td class="gsc_a_c"><a                href="https://scholar.google.com.tw/scholar?oi=bibs&amp;hl=en&amp;cites=3113880649589192398"                class="gsc_a_ac gs_
ibl">54</a></td>                <td class="gsc_a_y"><span class="gsc_a_h gsc_a_hc gs_ibl">2013</span></td>                </tr>                <tr class="gsc_a_tr">                <td c
lass="gsc_a_t"><a href="https://ieeexplore.ieee.org/abstract/document/6154073"                data-href="/citations?view_op=view_citation&amp;hl=en&amp;user=5o3YVKQAAAAJ&amp;sortby=pu
bdate&amp;citation_for_view=5o3YVKQAAAAJ:SeFeTyx0c_EC"                class="gsc_a_at">Human action recognition using histogram of oriented gradient of motion history
    image</a>                <div class="gs_gray">CP Huang, CH Hsieh, KT Lai, WY Huang</div>                <div class="gs_gray">Instrumentation, Measurement, Computer, Communicati
on and Control, 2011?…<span                class="gs_oph">, 2011</span></div>                </td>                <td class="gsc_a_c"><a                href="https://scho
lar.google.com.tw/scholar?oi=bibs&amp;hl=en&amp;cites=4337304926624707491"                class="gsc_a_ac gs_ibl">39</a></td>                <td class="gsc_a_y"><span class="gsc_a_h
 gsc_a_hc gs_ibl">2011</span></td>                </tr>                <tr class="gsc_a_tr">                <td class="gsc_a_t"><a href="https://ieeexplore.ieee.org/abstract/document/6045
516"                data-href="/citations?view_op=view_citation&amp;hl=en&amp;user=5o3YVKQAAAAJ&amp;sortby=pubdate&amp;citation_for_view=5o3YVKQAAAAJ:ns9cj8rnVeAC"
  class="gsc_a_at">Orientation histogram of SIFT displacement for recognizing actions in broadcast                videos</a>                <div class="gs_gray">KT Lai, MS Chen, C
H Hsieh, MF Lai</div>                <div class="gs_gray">Visual Information Processing (EUVIP), 2011 3rd European Workshop on,                286-291<span class="gs_oph">, 2011</
span></div>                </td>                <td class="gsc_a_c"><a                href="https://scholar.google.com.tw/scholar?oi=bibs&amp;hl=en&amp;cites=14667619654157581438"
                class="gsc_a_ac gs_ibl">3</a></td>                <td class="gsc_a_y"><span class="gsc_a_h gsc_a_hc gs_ibl">2011</span></td>                </tr>                <tr
class="gsc_a_tr">                <td class="gsc_a_t"><a href="https://link.springer.com/chapter/10.1007/978-3-642-13681-8_51"                data-href="/citations?view_op=view_citat
ion&amp;hl=en&amp;user=5o3YVKQAAAAJ&amp;sortby=pubdate&amp;citation_for_view=5o3YVKQAAAAJ:R3hNpaxXUhUC"                class="gsc_a_at">Human action recognition using key points displ
acement</a>                <div class="gs_gray">KT Lai, CH Hsieh, MF Lai, MS Chen</div>                <div class="gs_gray">International Conference on Image and Signal Processing,
439-447<span                class="gs_oph">, 2010</span></div>                </td>                <td class="gsc_a_c"><a                href="https://scholar.google.com.t
w/scholar?oi=bibs&amp;hl=en&amp;cites=7086163017893413872"                class="gsc_a_ac gs_ibl">8</a></td>                <td class="gsc_a_y"><span class="gsc_a_h gsc_a_hc gs_ibl"
>2010</span></td>                </tr>                <tr class="gsc_a_tr">                <td class="gsc_a_t"><a href="https://link.springer.com/chapter/10.1007/978-3-540-92957-4_66"
                data-href="/citations?view_op=view_citation&amp;hl=en&amp;user=5o3YVKQAAAAJ&amp;sortby=pubdate&amp;citation_for_view=5o3YVKQAAAAJ:hFOr9nPyWt4C"                class=
"gsc_a_at">Player detection and tracking in broadcast tennis video</a>                <div class="gs_gray">YC Jiang, KT Lai, CH Hsieh, MF Lai</div>                <div class="gs_gra
y">Pacific-Rim Symposium on Image and Video Technology, 759-770<span                class="gs_oph">, 2009</span></div>                </td>                <td class="gsc_a_c"><a
                href="https://scholar.google.com.tw/scholar?oi=bibs&amp;hl=en&amp;cites=17062666776910636 9"                class="gsc_a_ac gs_ibl">13</a>                </td>
                <td class="gsc_a_y"><span class="gsc_a_h gsc_a_hc gs_ibl">2009</span>                </td>                </tr>                </tbody>                </table>                </div>
        </div>        </div>    </section>    <br />    <br />    <!--==========================        Call To Action Section    ==========================-->    <!--    <section id="call-to-
action" class="wow fadeInUp">        <div class="container">            <div class="row">                <div class="col-lg-9 text-center text-lg-left">                <h3 class="cta-title">Call To Action<
/h3>                <p class="cta-text"> Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, s
unt in culpa qui officia deserunt mollit anim id est laborum.</p>                </div>                <div class="col-lg-3 cta-btn-container text-center">                <a class="cta-btn align-middle" hr
ef="#">Call To Action</a>                </div>            </div>        </div>    </section>    -->    <!--==========================        Contact Section    ==========================-->    <sect
ion id="contact" class="wow fadeInUp">        <div class="container">            <div class="section-header">                <h2>Contact Us</h2>                <p></p>                </div>            <div class="row co
ntact-info">                <div class="col-md-4">                <div class="contact-address">                <i class="ion-ios-location-outline"></i>                <h3>Office Address</h3>
  <address>Complex Building (綜科館), Room 406-1. No. 1, Sec. 3, Zhongxiao E. Rd., Taipei City, 10643, Taiwan                (臺北科技大學)</address>                </div>                </div>
    <div class="col-md-4">                <div class="contact-phone">                <i class="ion-ios-telephone-outline"></i>                <h3>Phone Number</h3>                <p>(+886-2) 2771-2171
 Ext. 2275</p>                </div>                </div>                <div class="col-md-4">                <div class="contact-email">                <i class="ion-ios-email-outline"></i>                <h3>E
mail</h3>                <p><a href="mailto:kuantinglai@gmail.com">ktlai<img border="0" src="img/at-teken.png">ntut.edu.tw</a></p>                </div>                </div>                </div>
    </div>    </section><!-- #contact -->  </main>  <!--==========================    Footer    ==========================-->  <footer id="footer">    <div class="container">        <div cla
ss="copyright">            &copy; Copyright <strong>Reveal</strong>. All Rights Reserved        </div>        <div class="credits">        <!--            All the links in the footer should remain in
tact.            You can delete the links only if you purchased the pro version.            Licensing information: https://bootstrapmade.com/license/            Purchase the pro version with work
ing PHP/AJAX contact form: https://bootstrapmade.com/buy/?theme=Reveal        -->        Designed by <a href="https://bootstrapmade.com/">BootstrapMade</a>        </div>    </div>  </footer><
!-- #footer -->  <a href="#" class="back-to-top"><i class="fa fa-chevron-up"></i></a>  <!-- JavaScript Libraries --> <script src="lib/jquery/jquery.min.js"></script>  <script src="lib/jq
uery/jquery-migrate.min.js"></script>  <script src="lib/bootstrap/js/bootstrap.bundle.min.js"></script>  <script src="lib/easing/easing.min.js"></script>  <script src="lib/superfish/hoverIn
tent.js"></script>  <script src="lib/superfish/superfish.min.js"></script>  <script src="lib/wow/wow.min.js"></script>  <script src="lib/owlcarousel/owl.carousel.min.js"></script>  <script
src="lib/magnific-popup/magnific-popup.min.js"></script>  <script src="lib/sticky/sticky.js"></script>  <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyD8HeI8o-clNppZA-92oYlX
akhDPYR7XMY"></script>  <!-- Contact Form JavaScript File -->  <script src="contactform/contactform.js"></script>  <!-- Template Main Javascript File -->  <script src="js/main.js"></script>
</body></html>
PS C:\Users\kuant\OneDrive\Teaching\108-2物件導向程式設計\code>
```

# InetAddress Class

```java
import java.io.*;
import java.net.*;
public class InetDemo{
    public static void main(String[] args){
        try{
        InetAddress ip=InetAddress.getByName("www.aiotlab.org");

        System.out.println("Host Name: "+ip.getHostName());
        System.out.println("IP Address: "+ip.getHostAddress());
        }catch(Exception e){ System.out.println(e);}
    }
}
```



# UDP: DatagramSocket class

- Creating a UDP server

```
import java.net.*;
public class UDPReceiver{
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket(3000);
        byte[] buf = new byte[1024];
        DatagramPacket dp = new DatagramPacket(buf, 1024);
        ds.receive(dp);
        String str = new String(dp.getData(), 0, dp.getLength());
        System.out.println(str);
        ds.close();
    }
}
```

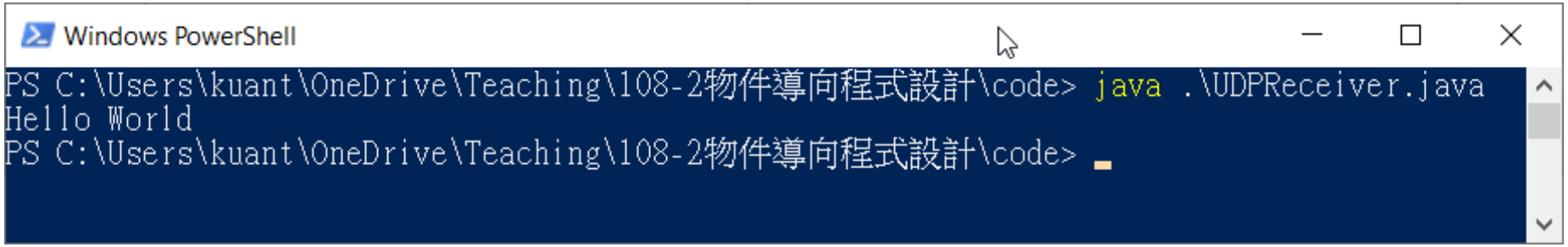
# Sending DatagramPacket

- Sending a packet to localhost “127.0.0.1”

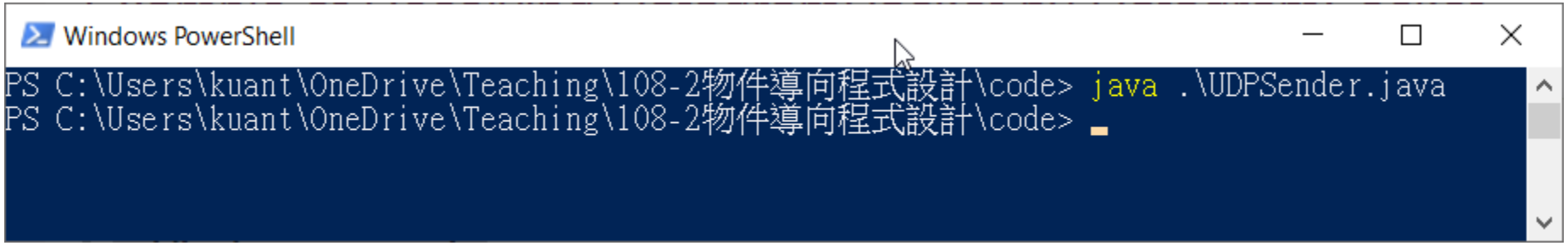
```
import java.net.*;
public class UDPSender{
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket();
        String str = "Hello World";
        InetAddress ip = InetAddress.getByName("127.0.0.1");

        DatagramPacket dp = new DatagramPacket(str.getBytes(), str.length(), ip, 3000);
        ds.send(dp);
        ds.close();
    }
}
```

# Test UDP server and client

- Run the server first

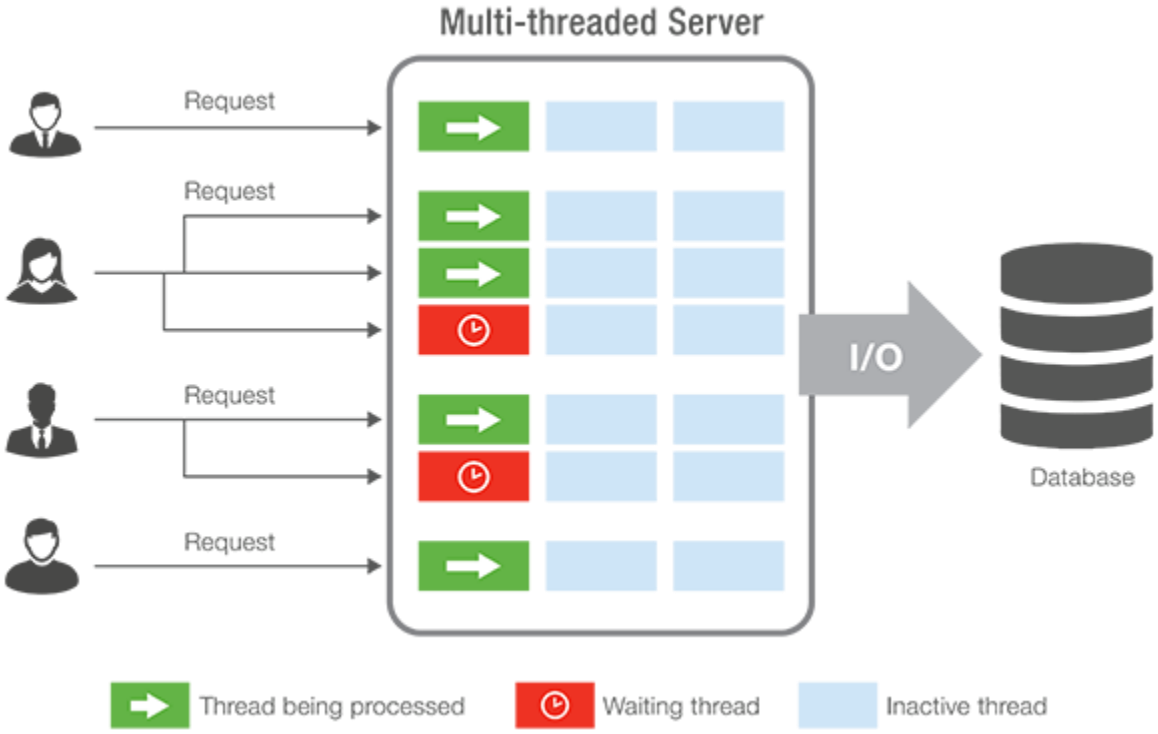



- Send packet using client



# Multithread TCP Server

- Previous example can handle only one client
- For multiple clients: using threads
  - Handle one client in one thread




https://www.eginnovations.com/blog/java-threads/

# Building a Multithread TCP Server

1. Create a `ServerSocket` and specify a port to listen on
2. Invoke the `ServerSocket`'s `accept()`to listen
3. When a client connects to the server, the `accept()` method returns a `Socket` through which the server can communicate with the client.
4. Pass the `Socket` to another thread to process so that your server can continue listening for additional connections.
5. Call the `ServerSocket`'s `accept()` method again to listen for another connection.

https://www.javaworld.com/article/2853780/socket-programming-for-scalable-systems.html

# Multithread Echo Server

```
public class SimpleSocketServer extends Thread
{
    private ServerSocket serverSocket;
    private int port;
    private boolean running = false;
    public SimpleSocketServer( int port ) { this.port = port;  }
    public void startServer() {
        try {
            serverSocket = new ServerSocket( port );
            this.start();
        }
        catch (IOException e) { e.printStackTrace(); }
    }
    public void stopServer() {
        running = false;
        this.interrupt();
    }
    @Override
    public void run() {
        ……
    }
    ……
```

https://www.javaworld.com/article/2853780/socket-programming-for-scalable-systems.html

```java
public class SimpleSocketServer extends Thread
{
    private ServerSocket serverSocket;
    private int port;
    private boolean running = false;
    public SimpleSocketServer( int port ) { this.port = port;  }
    ……
    ……
    @Override
    public void run() {
        running = true;
        while( running ) {
            try {
                System.out.println( "Listening for a connection" );
                Socket socket = serverSocket.accept();
                // Pass the socket to the RequestHandler thread for processing
                RequestHandler requestHandler = new RequestHandler( socket );
                requestHandler.start();
            }
            catch (IOException e) { e.printStackTrace();}
        }
    }
    ……
```

```
public class SimpleSocketServer extends Thread
{
    private ServerSocket serverSocket;
    private int port;
    private boolean running = false;
    public SimpleSocketServer( int port ) { this.port = port;  }
    ……
    ……
    ……
    public static void main( String[] args )
    {
        if( args.length == 0 ) {
            System.out.println( "Usage: SimpleSocketServer <port>" );
            System.exit( 0 );
        }
        int port = Integer.parseInt( args[ 0 ] );
        System.out.println( "Start server on port: " + port );
        SimpleSocketServer server = new SimpleSocketServer( port );
        server.startServer();
    }
}
```

# Handle Client Request

```
class RequestHandler extends Thread {
    private Socket socket;
    RequestHandler( Socket socket ) { this.socket = socket; }
    @Override
    public void run() {
        try {
            System.out.println( "Received a connection" );
            BufferedReader in = new BufferedReader( new InputStreamReader(socket.getInputStream()) );
            PrintWriter out = new PrintWriter( socket.getOutputStream() );
            out.println( "Echo Server 1.0" );
            out.flush();
            String line = in.readLine();
            while( line != null && line.length() > 0 )
            {
                out.println( "Echo: " + line );
                out.flush();
                line = in.readLine();
            }
            // Close our connection
            in.close();
            out.close();
            socket.close();
            System.out.println( "Connection closed" );
        }
        catch( Exception e ) { e.printStackTrace();}
    }
}
```

# Reference


1. https://www.javatpoint.com/socket-programming
2. https://en.wikipedia.org/wiki/Internet_protocol_suite
3. http://mhshohag.com/compare-and-contrast-osi-and-tcp-ip-models/
4. https://sites.google.com/site/yutbms/osi-model