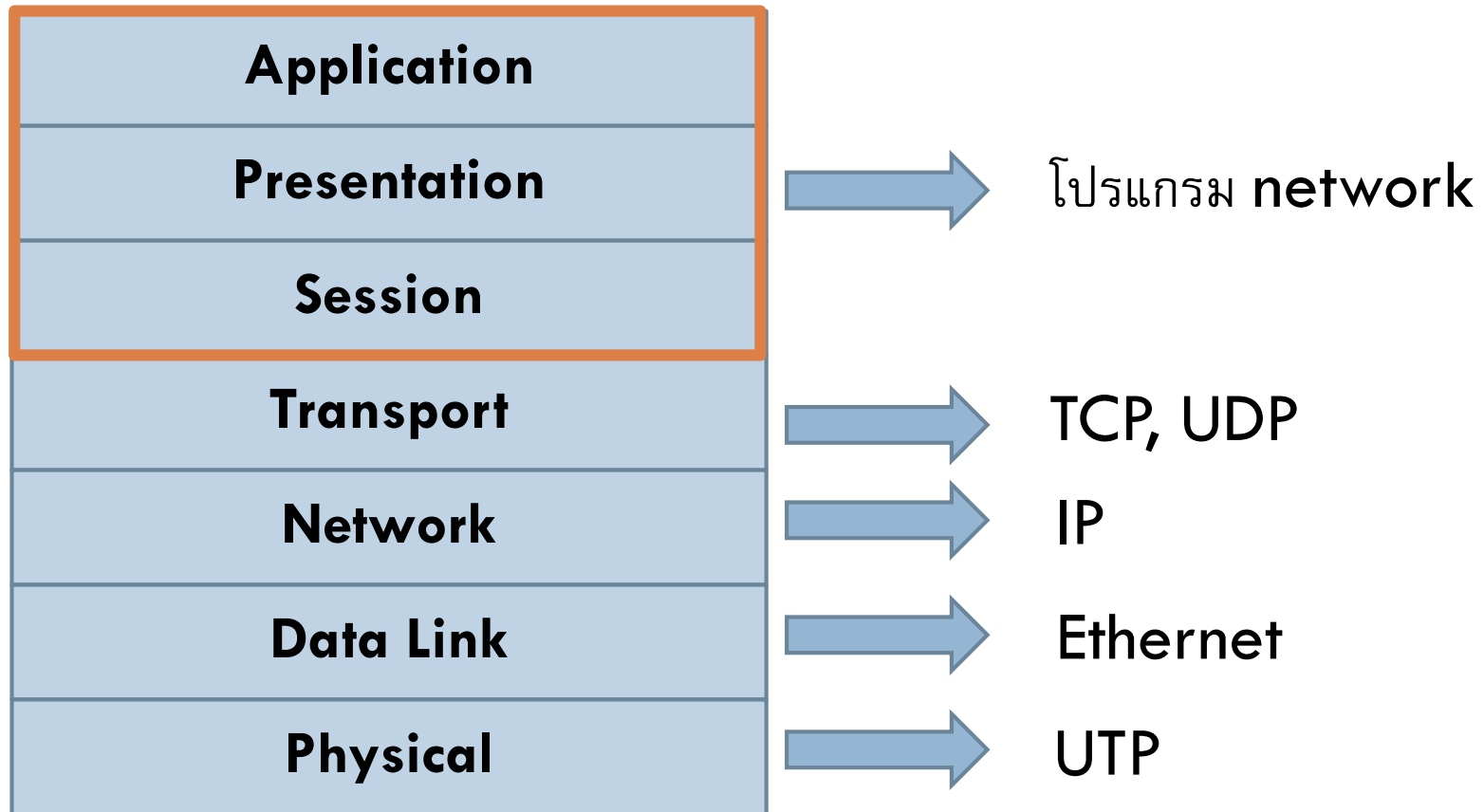


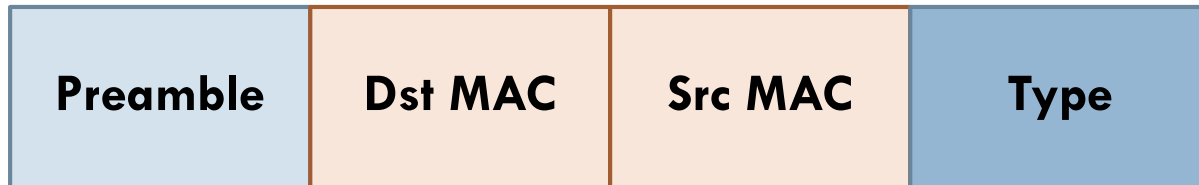
LOOKING UP IP

030523313 - Network programming
Asst. Prof. Dr. Choopan Rattanapoka

ทบทวน OSI layer



Ethernet Header

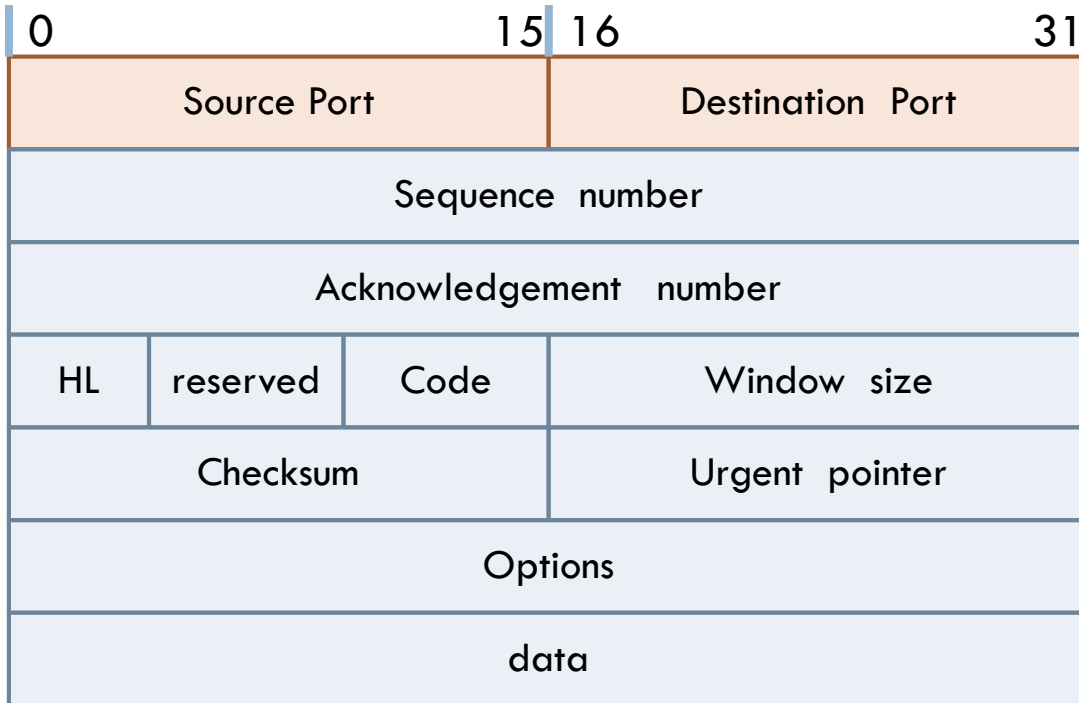


- **Preamble** (8 bytes)
- **Dst MAC** : MAC address ของเครื่องปลายทาง (6 bytes)
- **Src MAC** : MAC address ของเครื่องต้นทาง (6 bytes)
- **Type** (2 bytes) : ใช้บอก protocol ของชั้น network
 - 0x0800 Internet Protocol, Version 4 ([IPv4](#))
 - 0x0806 Address Resolution Protocol ([ARP](#))
 - 0x8035 Reverse Address Resolution Protocol ([RARP](#))
 - 0x8137 Novell [IPX](#) (alt)
 - 0x86DD Internet Protocol, Version 6 ([IPv6](#))

IP header

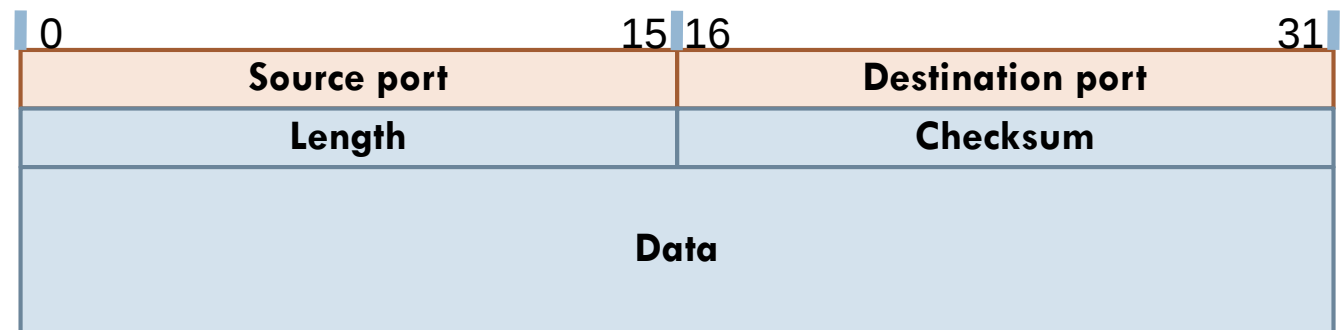
0		15		16		31	
Version	IHL	TOS		Total Length			
Identification				Flags	Fragment Offset		
Time to Live		Protocol		Header checksum			
Source IP address							
Destination IP address							
Options				Padding			
Data							
.							
.							

TCP and UDP Header

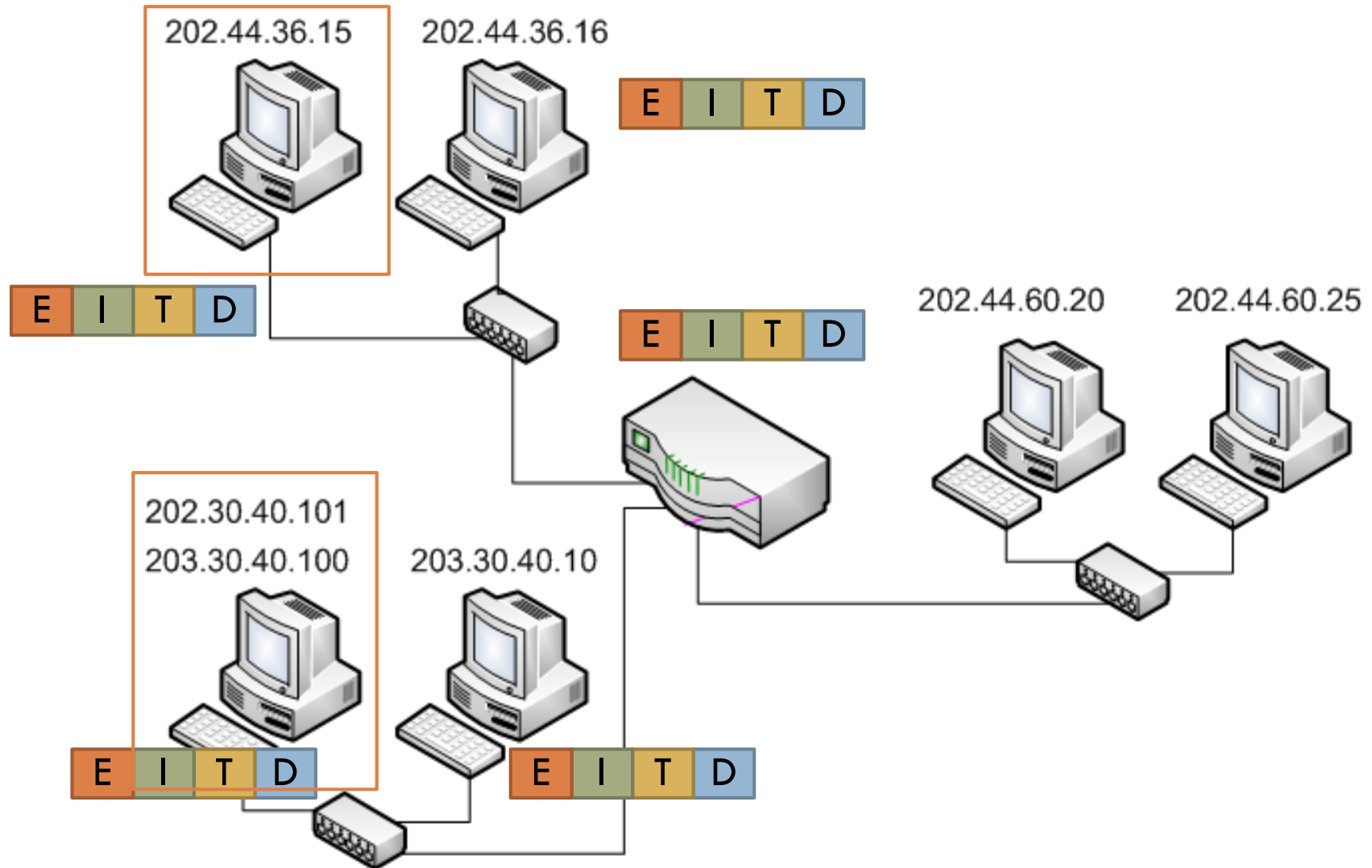


TCP header

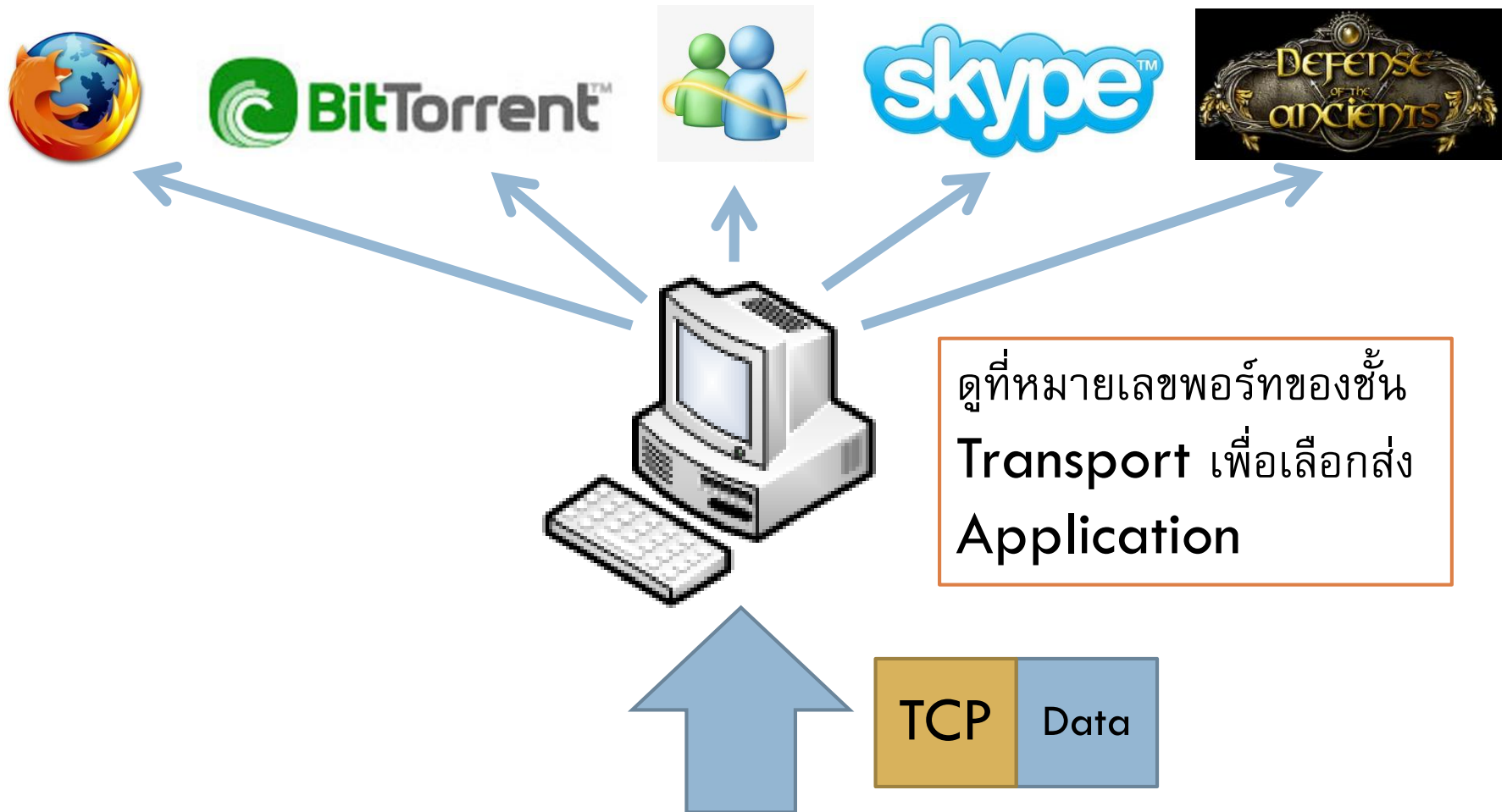
UDP header



ตัวอย่างการส่งข้อมูลในระบบเครือข่าย



การใช้งานของพอร์ต



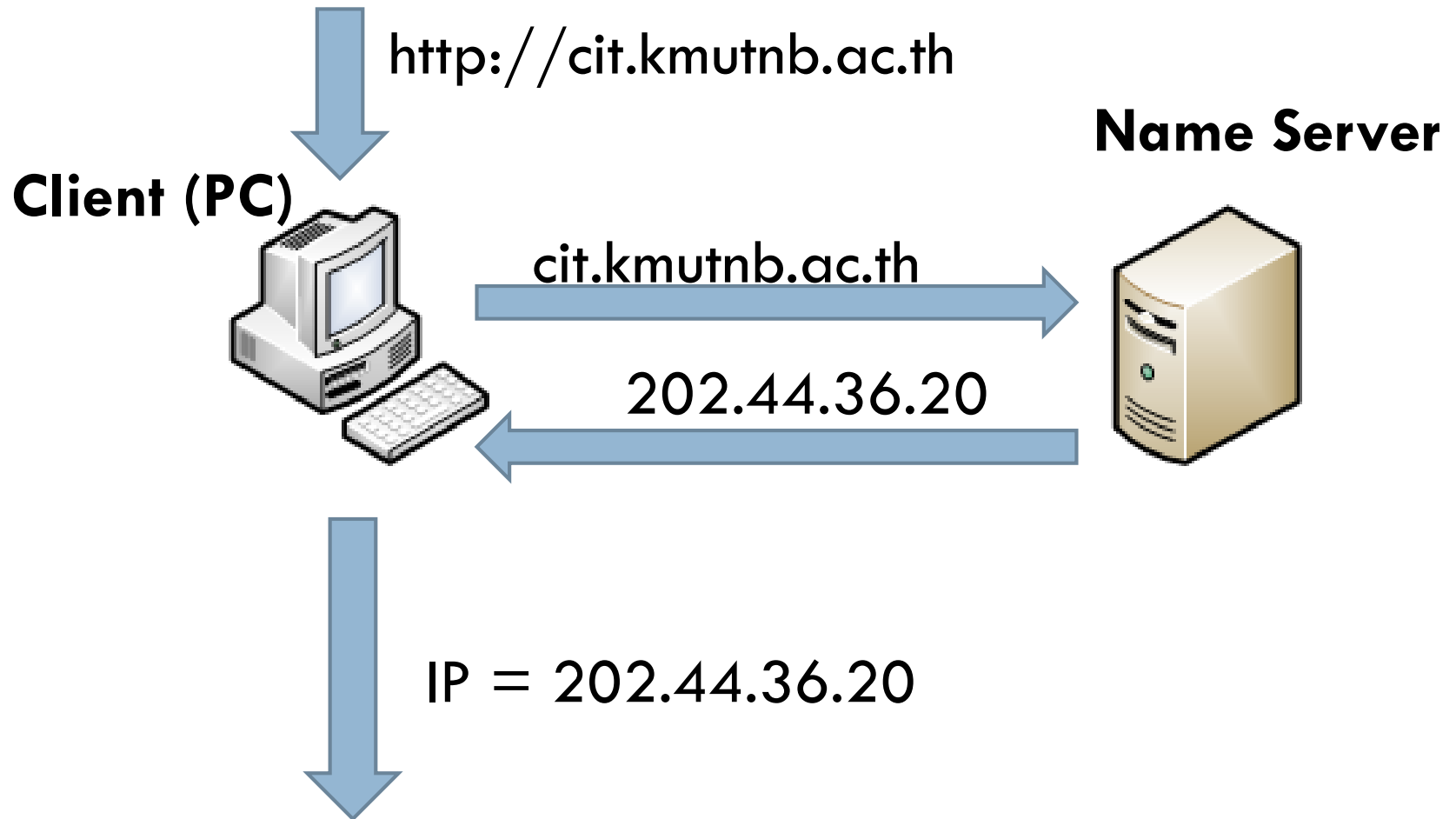
หมายเลขพอร์ต

- Port จะเป็นจำนวนเต็มขนาด 16 bits (1-65535)
- Port หมายเลข 1 – 1023 ถูกสงวนไว้เพื่อบริการของระบบ เรียกว่า “well-known ports”
- ตัวอย่างของ Port 1-1023 ที่ถูกสงวนไว้เพื่อบริการต่างๆ เช่น
 - Port 20,21 : FTP (File Transfer Protocol)
 - Port 23 : Telnet
 - Port 25 : SMTP (Simple Mail Transfer Protocol)
 - Port 53 : Domain
 - Port 80 : HTTP (HyperText Transfer Protocol)
 - Port 110 : POP3 (Post Office Protocol version 3)
- Well-known ports ถูกกำหนดไว้ในไฟล์
 - Windows -> C:\WINDOWS\system32\drivers\etc
 - Linux -> /etc/service

IP Address

- **IP Address** เปรียบเสมือนเลขที่บ้านของเครื่องคอมพิวเตอร์
- เมื่อมีการเชื่อมต่อกันระหว่างเครื่องจำเป็นต้องใช้หมายเลข **IP** เพื่อระบุตัวตน
- ในทางปฏิบัติการจำหมายเลข **IP** เพื่อติดต่อไปเครื่องอื่นนั้นไม่สะดวก จึงมีบริการที่ชื่อ **DNS (Domain name system)** เข้ามาช่วย
- **DNS** ทำหน้าที่แปลงจากชื่อ **host** ให้เป็น **IP** และแปลงจากหมายเลข **IP** ให้เป็นชื่อ **host**

การทำงานของ DNS



การหาหมายเลข IP ด้วยภาษา Java

- จะใช้ class `InetAddress` ซึ่งมี static เมธอดคือ
 - ▣ `public static InetAddress getByName(String hostname) throws UnknownHostException`
 - คื object ของ `InetAddress` ที่เก็บข้อมูลของ `hostname`
 - ▣ `public static InetAddress[] getAllByName(String hostname) throws UnknownHostException`
 - คื array object ของ `InetAddress` ที่เก็บข้อมูลของ `hostname`
 - ▣ `public static InetAddress getLocalHost( ) throws UnknownHostException`
 - คื object ของ `InetAddress` ที่เก็บข้อมูลของ `localhost`

InetAddress

- ใน `class InetAddress` เองมีเมธอดที่ใช้ดึงข้อมูลออกมาคือ
 - ▣ `public String getHostName()`
 - ใช้สำหรับดึงชื่อ host
 - ▣ `public byte[ ] getAddress()`
 - ใช้สำหรับดึง IP address ซึ่งอยู่ในรูปของ byte array
 - ▣ `public String getHostAddress()`
 - ใช้สำหรับดึง IP address ในรูปของ String

ตัวอย่าง 1

```
import java.net.*;

public class TestInet {

    public static void main(String[ ] args) {

        try {

            InetAddress ad = InetAddress.getByName("ect.cit.kmutnb.ac.th");

            System.out.println("IP = " + ad.getHostAddress());

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

}
```

ตัวอย่าง 2

```
import java.net.*;
public class TestInet2 {
    public static void main(String[ ] args) {
        try {
            InetAddress[ ] ad =
                InetAddress.getAllByName("www.google.com");
            for(int i = 0; i < ad.length; i++) {
                System.out.println("IP = " + ad[i].getHostAddress());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

ตัวอย่าง 3

```
import java.net.*;
public class TestInet2 {
    public static void main(String[ ] args) {
        try {
            InetAddress ad = InetAddress.getLocalHost();
            System.out.println("Host = " + ad.getHostName());
            System.out.println("IP = " + ad.getHostAddress());
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

ตัวอย่าง 4

```
import java.net.*;
public class TestInet {
    public static void main(String[ ] args) {
        try {
            InetAddress ad = InetAddress.getByName("202.44.32.13");
            System.out.println("host = " + ad.getHostName());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Socket

- **Socket** หรือ **Socket Address** หมายถึง คู่ของ **IP Address** กับ **Port**
- เครื่องแต่ละเครื่องจะมี **IP** ไม่ซ้ำกัน
- ในเครื่องหนึ่งเครื่องแต่ละ **Application** จะใช้พอร์ตไม่ซ้ำกัน
- **Socket** เมื่อมองในแง่ของการเขียนโปรแกรม อาจมองได้ว่าเป็นท่อเชื่อมระหว่างเครื่อง 2 เครื่องที่จะติดต่อสื่อสารกัน

Class Socket

- ในภาษา Java มี Class ชื่อ Socket เพื่อทำหน้าที่เชื่อมต่อเครื่องคอมพิวเตอร์เข้าด้วยกัน โดยใช้ TCP
- Class Socket มี constructor อยู่หลายตัวแต่ตัวที่นิยมใช้คือ
public Socket(String host, int port) throws UnknownHostException, IOException
 - host ชื่อเครื่องหรือหมายเลข IP ของเครื่องปลายทาง
 - port หมายเลขพอร์ตของเครื่องปลายทาง
 - UnknownHostException ถ้า host ที่ใส่ไม่มีอยู่จริง
 - IOException เมื่อมีปัญหาในการเชื่อมต่อ

ตัวอย่างการเชื่อมต่อเครื่องปลายทาง

```
1  import java.io.*;
2  import java.net.*;
3
4  public class TestJava {
5      public static void main(String[] args) {
6          try {
7              Socket socket = new Socket("127.0.0.1", 80);
8          }
9          catch (UnknownHostException unknownEx) {
10             unknownEx.printStackTrace();
11          }
12          catch (IOException ioEx) {
13             ioEx.printStackTrace();
14          }
15      }
16  }
```

ดักจับ Exception

```
1  import java.io.*;
2  import java.net.*;
3
4  public class TestJava {
5      public static void main(String[] args) {
6          try {
7              Socket socket = new Socket("127.0.0.1", 80);
8          }
9          catch (Exception e) {
10             e.printStackTrace();
11          }
12     }
13 }
```